AFRL-HE-WP-TR-2004-0072

United States Air Force
Research Laboratory

AIE CSEDS: Initial Cognitive Systems Engineering Design Specification (CSEDS) for the ACWA™ Integrated Environment (AIE)

James Gualtieri
Scott Potter
William Elm
Jack McKee

ManTech Aegis Research Corporation
Cognitive Systems Engineering Center
502 Grant Street, Suite 475
Pittsburgh, PA 15219

June 2003

Final Report for the Period November 2001 to June 2003

Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

When US Government drawings, specifications or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., Ste 0944
Ft. Belvoir, VA 22060-6218

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2004-0072

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

// Signed //

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2003	Final Report - November 2001 - June 2003

4. TITLE AND SUBTITLE
AIE CSEDS: Initial Cognitive Systems Engineering Design Specification (CSEDS) for the ACWA™ Integrated Environment (AIE)

5. FUNDING NUMBERS
C: F33601-02-F-A219
PE: 63231F
PR: 2830
TA: 30
WU: 03

6. AUTHOR(S)
James Gualtieri, Scott Potter, William Elm, Jack McKee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ManTech Aegis Research Corporation
Cognitive Systems Engineering Center
502 Grant Street, Suite 475
Pittsburgh, PA 15219

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory, Human Effectiveness Directorate
Warfighter Interface Division
Air Force Materiel Command
Cognitive Systems Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-2004-0072

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This project is focused on developing an Integrated Development Environment (IDE) as a means for supporting a particular Cognitive Systems Engineering (CSE) methodology. This document examines the capabilities and features necessary to support the Applied Cognitive Work Analysis (ACWA™) methodology. Such an IDE would enable decision support systems (DSS) in the areas of information warfare, command and control, and other complex military (as well as commercial) applications to be developed based on the cognitive demands of the domain. In addition, an IDE will enable these systems to be developed more quickly, at a lower cost, and with greater functionality than is currently possible using the available manually intensive tools. This project seeks to specify both the overall shape of the proposed IDE, as well as the specific displays. The envisioned IDE will be referred to herein as the ACWA Integrated Environment, or AIE, and will serve as the integration point between insights gained from the ACWA™ design artifacts and the resulting software engineering development activities. AIE will be a tool for software developers to maintain awareness of the "design basis" underlying the resulting system requirements and specifications, by forming a maintainable, traceable component of the functional design. Thus, our vision is to develop an IDE that simultaneously and seamlessly aids the experienced CSE analyst in the modeling and documentation aspects of the ACWA™ process, and the equally experienced software developer, during the construction of the resulting DSS.

14. SUBJECT TERMS
Cognitive Systems Engineering Integrated Development Environment
Information Warfare Applied Cognitive Work Analysis (ACWA™)
Decision Support Systems

15. NUMBER OF PAGES
52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This project is focused on developing an Integrated Development Environment (IDE) as a means for supporting a particular Cognitive Systems Engineering (CSE) methodology. This document examines the capabilities and features necessary to support the Applied Cognitive Work Analysis (ACWA™) methodology. Such an IDE would enable decision support systems (DSS) in the areas of information warfare, command and control, and other complex military (as well as commercial) applications to be developed based on the cognitive demands of the domain. In addition, an IDE will enable these systems to be developed more quickly, at a lower cost, and with greater functionality than is currently possible using the available manually intensive tools. The effort described in the report is only an initial attempt to develop a complete design specification to support cognitive systems engineers.

This project seeks to specify both the overall shape of the proposed IDE, as well as the specific displays. The envisioned IDE will be referred to herein as the ACWA Integrated Environment, or AIE, and will serve as the integration point between insights gained from the ACWA™ design artifacts and the resulting software engineering development activities. AIE will be a tool for software developers to maintain awareness of the "design basis" underlying the resulting system requirements and specifications, by forming a maintainable, traceable component of the functional design. Thus, our vision is to develop an IDE that simultaneously and seamlessly aids the experienced CSE analyst in the modeling and documentation aspects of the ACWA™ process, and the equally experienced software developer, during the construction of the resulting DSS.

The approach used for developing AIE used CSE roles to define the necessary system functions. However, rather than use of roles to define how users interact with the system, roles, in this case, are used to identify the types of tasks that a user in that role would need to accomplish as well as the products that they need to produce. This framework served as a guide when identifying the components necessary for inclusion in the AIE system.

For this design effort ManTech Aegis Research Corporation has chosen not to place any constraints on the size of the workspace. By designing using an unconstrained workspace, some of the serialism and parallelism issues are left for future design refinements. The AIE workspace is envisioned to be composed of twelve (12) panes. As an IDE, each of the panes in the workspace design of AIE is potentially impacted by actions/modifications in any other pane. The detailed descriptions of the individual panes that comprise AIE were organized into three subsections/categories: (1) Analysis, (2) Design, and (3) Management functions.

After the discussion of the system components, this document contains an operational scenario for how this system would be used by a cognitive systems engineer. The purpose of the operational scenario is to convey the overarching vision of the proposed system to fulfill the decision support needs of the cognitive engineers. This will include an overview of the envisioned system and a discussion of the major system (display) components. This allows an understanding of not only the details, but also the overarching intent of the AIE system.

The next steps in the development of the ACWA Integrated Environment (AIE) system should be to extend what was learned during the technical investigation of Integrated Development Environments (IDE) and use the preliminary Cognitive Systems Engineering Design Specification (CSEDS) to construct a limited scope AIE prototype. The AIE prototype will focus on developing support for the Analytical portion of Cognitive Systems Engineering (CSE), which is the heart of a fully functioning AIE system. Constructing this prototype will enable AFRL to better achieve its mission of efficiently develop decision support systems from a decision-centered perspective.

A INTRODUCTION

A.1 Purpose of This Report

The goal of this report is to begin to ***define the requirements for the development of an Integrated Development Environment (IDE) to support Cognitive Systems Engineering (CSE), in order to build highly effective decision support systems (DSS)***. Such an IDE would enable DSS in the areas of information warfare, command and control, and other complex military (as well as commercial) applications to be developed based on the cognitive demands of the domain. In addition, an IDE will enable these systems to be developed more quickly, at a lower cost, and with greater functionality than is currently possible using the available manually intensive tools. The effort described in the report is only an initial attempt to develop a complete design specification to support cognitive systems engineers.

A.2 Scope of This Project

This project is focused on developing an IDE as a means for supporting the CSE methodology. The current effort consists of two components: a technical evaluation and a requirement specification. The technical evaluation will efficiently capture the results of a technical survey. Based on the results of this survey, decisions can be made about tools, products, etc., that will form components of an IDE's implementation.

The requirement specification addressed in this document examines the capabilities and features necessary to support the Applied Cognitive Work Analysis (ACWA™) methodology. The requirement specification for a CSE IDE will be made based on an examination of a cognitive systems engineer's functions, as well as an examination of leading edge software IDE.

This work effort is focused on taking the next step in the development of an IDE in support of cognitive systems engineers. ***This project seeks to specify both the overall shape of the proposed IDE, as well as the specific displays.***

Based on the outcome of the requirement specification effort, recommendations will be made for "next steps" in the system's development. ***The goal of this development is to produce an effective computer based design aid that is a practical tool for system designers through out the Air Force, as well as for their supporting contractors.***

A.3 Scope of This Document

A.3.1 Contents of this Report

This report is intended to convey the essential characteristics of the envisioned IDE. This report is a tailored version of the ManTech Aegis Research Corporation's standard Cognitive Systems Engineering Design Specification (CSEDS). Some sections have been omitted, because no formal analysis of a cognitive systems engineer was completed as part of this effort.

The envisioned IDE will be referred to herein as the Applied Cognitive Work Analysis Integrated Environment, or AIE, and will serve as the integration point between insights gained from the ACWA™ design artifacts and the resulting software engineering development activities. This is expected to result in a seamless development environment, from fundamental domain structure and cognitive demands, through software design artifacts, to the resulting implementation of the DSS.

This report contains four major sections. Section B contains an operational scenario describing AIE. This scenario touches on how a cognitive systems engineer would interact with the envisioned system.

Section B provides an overview of the proposed IDE to support cognitive systems engineers. This section also specifies the set of displays necessary to support CSE functions.

Section D is a detailed description of each of the components that comprise the proposed AIE system. This section also provides an initial set of requirements for each display.

Section E focuses on future development phases for the AIE system. This section discusses the transition path from limited capability prototype to fully functioning system.

A.3.2 Relating this Report to Prior and Future Work

This report continues the effort begun over five years ago to develop a tool to support cognitive systems engineers and enable greater integration into the software/system development process. This effort started with the development of a research prototype Computer-Aided Cognitive Systems Engineer (CACSE) design tool. CACSE was developed to test the practicality of computerized assistance to decision support system designers using a Cognitive Work Analysis-based approach to develop decision aids. The prototype also explored the various engineering issues that would be required from a full design tool.

The evaluation of the CACSE prototype that occurred last year reinforced the value of a support environment for experienced Applied Cognitive Work Analysis (ACWA™) practitioners. However, as with any prototype, its "lessons learned" provide guidance for the construction of a fully functional system in order for it to be a practical tool for use by system designers across the Air Force, as well as for their supporting contractors.

In this report, a first attempt is made at developing a requirements document for a fully functional software system to support cognitive system engineers using the ACWA™ approach. Due to the scope of this effort, the identification of requirements was conducted based on an overview of the CSE process, rather than by the more rigorous functional analysis of a CSE practitioner. Therefore, ***the requirements contained within this document are tentative, and should be validated in future related efforts***. Specific future steps are for the development of an IDE for CSE is discussed in Section E.

A.4 Release / Classification Information

A.4.1 Security Classification

This report has been determined to be UNCLASSIFIED.

A.4.2 Release / Distribution Restrictions

None, Unlimited Distribution

A.5 References

1. Aegis Research Corp., Cognitive Systems Engineering Center (2001). Distributed crew interface for autonomous satellite operations (PO # F33615-01-M-6023). Small Business and Innovative Research (SBIR) for Phase I Interim Report for Air Force Research Laboratory. Wright-Patterson AFB, OH: Crew Systems Ergonomics Information Analysis Center.
2. Booch, Rumbaugh, & Jacobson (1998). The Unified Modeling Language User Guide. Addison-Wesley, 1998.

3. Coad, P & Mayfield, M. (1997). Java Design: Building Better Apps & Applets (2nd Ed.). Prentice Hall.
4. Elm, W.C., Potter, S.S., Gualtieri, J.G., Roth, E.M., & Easter, J.R. (May, 2003). Applied cognitive work analysis: A pragmatic methodology for designing revolutionary cognitive affordances. In E. Hollnagel (Ed.), Handbook of Cognitive Task Design. London: Lawrence Erlbaum Associates, Inc.
5. Fowler, M (1997). UML Distilled: Applying the Standard Object Modeling Language. Addison-Wesley.
6. Gualtieri, J.G., Roth, E.M., & Elm, W.C. (1999). Cognitive task analysis of teams: Utilizing cognitive science engineering for managing team coordination complexity. Small Business and Innovative Research (SBIR) for Phase I Final Report for Air Force Research Laboratory. Wright-Patterson AFB, OH: Crew Systems Ergonomics Information Analysis Center.
7. Java 2 Enterprise Edition versus The .NET Platform Two Visions for eBusiness by Roger Session (http://www.objectwatch.com/FinalJ2EEandDotNet.doc)
8. Lind, M. (1991). Representations and abstractions for interface design using multilevel flow modeling. In G. R. S. Weir and J. L. Alty (Eds.), Human-Computer Interaction and Complex Systems, London Academic Press.
9. Lind, M. (1993). Multilevel flow modeling. AAI93 Workshop on Reasoning about Function, July 11-15, Washington, DC.
10. ManTech Aegis Research Corp., Cognitive Systems Engineering Center (2002). AIE: ACWA™ Integrated Environment Tool Software Technical Survey (PO # F33601-02-F-A219). Final Report for Air Force Research Laboratory. Wright-Patterson AFB, OH: Crew Systems Ergonomics Information Analysis Center.
11. Potter, S. S., Ball, R. W., Jr., and Elm, W. C. (Aug., 1996). Supporting aeromedical evacuation planning through information visualization. In Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems. Dayton, OH: IEEE. pp. 208-215.
12. Potter, S. S., Roth, E. M., Woods, D. D. & Elm, W. (2000). Bootstrapping Multiple Converging Cognitive Task Analysis Techniques for System Design. In Schraagen, J.M., Chipman, S.F., & Shalin, V.L. (Eds.), Cognitive Task Analysis. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
13. Rasmussen, J. (1986). Information processing and human-machine interaction: An approach to cognitive engineering. North Holland Series in System Science and Engineering, Elsevier Science Publishing Co., Inc., New York, New York.
14. Rasmussen J., Pejtersen, A. M., & Goodstein, L. P. (1994). Cognitive Systems Engineering. New York: Wiley & Sons.
15. Roth, E. M. & Mumaw, R. J. (1995). Using Cognitive Task Analysis to Define Human Interface Requirements for First-of-a-Kind Systems. Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting, (pp. 520 – 524). Santa Monica, CA: Human Factors and Ergonomics Society.
16. Schraagen, J.M.C., Chipman, S.F., & Shalin, V.L. (Eds.) (2000). Cognitive Task Analysis. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
17. SOAP Version 1.2 Part 0: Primer (http://www.w3.org/TR/2001/WD-soap12-part0-20011217)
18. TogetherSoft (2002). Together® ControlCenter™ User Guide Version 6.0.1. (http://www.togethersoft.com/support)
19. Vicente, K. (1999). Cognitive Work Analysis. Mahwah, NJ: Lawrence Erlbaum Associates.
20. Vicente, K. and Rasmussen, J. (1992). Ecological interface design: Theoretical Foundations. IEEE Transactions on Systems Man and Cybernetics, 22, 589-606.

21. Wolter, R (2002). XML Web Services Basics. Microsoft® Corp. (http://msdn.microsoft.com/library/en-us/Dnwebsrv/html/webservbasics.asp)
22. Woods, D. D. (1984). Visual momentum: A concept to improve the cognitive coupling of person and computer. International Journal of Man-Machine Studies, 21: 229-244.
23. Woods, D. D. and Hollnagel, E. (1987). Mapping cognitive demands in complex problem-solving worlds. International Journal of Man-Machine Studies, 26, pp. 257-275.
24. Woods, D.D., Johannesen, L.J., Cook, R.I. and Sarter, N.B. (1994). State of the Art Report (SOAR): Behind Human Error: Cognitive Systems, Computers, and Hindsight. Wright-Patterson AFD, OH: Crew Systems Ergonomics Information Analysis Center.

B OPERATIONAL SCENARIO

B.1 Role of an Operational Scenario within ACWA™

From the perspective of the Applied Cognitive Work Analysis (ACWA™) based system design process, one of the logical next steps after mapping the end-user decision space (i.e., building the Functional Abstraction Network) and developing the associated Cognitive Work Requirements (CWRs) and the supporting Information Relationship Requirements (IRRs), is to define high-level intent of the envisioned joint cognitive system (human plus computer) concept.

The purpose of the operational scenario is to convey the overarching vision of the proposed system. This will focus on operational features that are to be provided, but without specifying the design details. The scenario should provide an example of how the envisioned system would be used by a typical user. The operational scenario will include an overview of the envisioned system and a discussion of the major system (display) components. This allows designers to understand not only the details, but also the overarching intent. By fully embracing this intent, designers can better execute the intent in the construction of the system.

B.2 Introduction and the Road to War

A scene in the not too distant future: LTC Cogman has been freshly assigned to AFRL/HE to put that Cognitive Psychology/Industrial Engineering degree to work, and lands squarely in the middle of a firestorm:

LTC Cogman finds himself in an interesting situation; a previously developed Cognitive Work Analysis (CWA) is being pulled off the shelf for some more work. In this case, functions that were previously out of scope are suddenly the focus of attention. That always tends to happen when budgets limit the work a priori, but then 'a situation' occurs that gets labeled 'operator error'

> *These incidents would have better been described as 'operator unable to overcome design limitations of the decision aid during a first-of-a-kind crisis' or 'operator-decision aid team error' at least.*

But in any case, the After Action Review was enough to free up some budget to complete the rest of the system design. In addition, LTC Cogman finds himself challenged not only by the normal design pressures, but also in attempting to "use someone else's CWA" as a starting point. This project will break some new ground in several areas: new support features for the beleaguered user, one of very few successful uses of someone else's CWA, and one of the first actual uses of the new ACWA™ Integrated Environment (AIE).

AIE was developed to explicitly support developing advanced software for decision support using a Cognitive Work Analysis based approach. Fortunately for LTC Cogman, the earlier CWA had been captured in an AIE design database as one of the system testing exercises done by the development team. AIE was modeled on the Integrated Development Environments now in common use by Software Architects, Software Engineers and Programmers since many of AIE's powerful design features have analogous counterparts in a CWA based Cognitive Systems Engineering (CSE) development process.

B.3 Getting Started...adding on to an existing analysis

B.3.1 Understandability of Previous Analysis

Before diving in and undoing potentially valuable previous analysis results, LTC Cogman focuses on understanding the knowledge model represented by the partial FAN from the previous effort. If he can understand the thinking that went into the model he is more likely to remain consistent as he extends it.

> *Previous CWA efforts have struggled at this point as each analyst's 'style' was wildly divergent from the others'. Even the use of language often creates confusion.*

Within AIE, the lessons contained in recent updates to the ACWA methodology have resulted in features and completeness checking that ensures the design rationale is captured at each level of granularity in the process. The designer must explain how the functional concept came to be, the language of the goal and process nodes, even the choice of commodity and its labeling language. By reading the FAN from 'top' down, he is able to understand each step 'down' from the context established by the nodes it supports. After reviewing the goal process nodes, he is able to localize a triangular region of the FAN that seems to surround the area he intends to work.

Focusing in on this region in the FAN window at the center of the screen, LTC Cogman picks one of the goal-process nodes. Immediately, the surrounding windows representing different views into the design such as Cognitive Work Requirements (CWRs), Information Relationship Resources (IRRs), and Commodity Dictionary highlight their respective content directly associated with the FAN node selection.

> *This inter-view synchronized selection would have occurred no matter which individual view LTC Cogman had focused on and selected an element from.*

By 'reading' across the windows, he is able to quickly understand the functional commodity's definition, the process model structure, the definition of goal success and how the various support-supported links coming to and leaving from this node and how they combine into a complete understanding of that portion of the ACWA analysis. This 'glancing back and forth' across windows is done almost subconsciously, his eyes jumping to the immediately available perspective that fits the needs of his understanding

After understanding the commodity definition, the process model begins to make sense, by glancing at its detail view, the text description created by its original designer, including update comments with time tags and version numbers from the various designs who have touched it since, make it clear why it is the way it is. A quick look at the associated CWRs reflects the decisions associated with this function. Selecting a particular transformation node symbol on the FAN results in a smaller highlighted region of CWRs in its window, only those CWRs directly related to that process node are selected.

Something about the highlighting across coordinated windows reflecting the various perspectives makes it easy to mentally integrate the various perspectives into an understanding the original CWA. In addition, it helps him understand that highlighted portion within the context of the same type of object immediately before and after it within that perspective's view. LTC Cogman is surprised how rapidly he is able to 'walk in the shoes of the original design team'. He now has identified where to start adding his new functions, and zooms in to that portion of the FAN.

B.3.2 Adding a New Goal Process Node to the Model

As LTC Cogman stares thoughtfully at the Knowledge Elicitation (KE) notes, he begins to frame a functional Goal Process Node in his mind. Clicking on the tools for the FAN window, he selects "create" and does the usual corner-drag-drop draw of a rectangle. Only in AIE its not just a rectangle, it immediately takes on the personality of a Goal Process Node:

- It is given a unique ID tag to assist in cross referencing,

- It is divided into two regions: Goal and Process,
- The Process region has the default process model and a Commodity, and
- The Goal region has a default label, clearly indicating that the fields need to be selected and instantiated by the designer.

More surprisingly, all the other windows leap into action, each inserting the default structures for CWRs, IRRs, etc., each automatically highlighted to reflect their association with the new node. LTC Cogman clicks on <Commodity 107> and types "Non-Lethal Combat Power". AIE responds with an audible complaint, and a message in the status window of "Commodity already exists, select MORE to see ACWA guidance on commodity uniqueness". The Commodity Dictionary window has both the newly typed entry and the conflicting entry highlighted. By clicking on the original entry, the FAN window immediately scrolls to that region of the model and highlights the Goal Process node. All the other windows of the display also update to reflect their perspectives on this portion of the model. After reading the various windows, LTC Cogman realizes he should have been more precise and renames this commodity "Psychological Pressure".

In renaming it, an idea takes form: "this might actually be modeled similarly to any other 'make pressure' function" he mutters to himself. He selects the default process model and selects a replacement from the Process Model templates shown along the edge of the FAN Toolbar. This "make pressure" template is immediately instantiated (a distinct copy of the template is generated) in the process region of the node, creating several interconnected process symbols, and more surprisingly, creating several support functions, linking them to the appropriate process nodes with similar additions to all the other windows (CWRs, IRRs etc.) Since it was a templated function, the CWRs and IRRs are instantiations of the more detailed ones from the template.

LTC Cogman very quickly has a lot of content going in this new portion of the ACWA model.

B.3.3 Hints of the Future

As the surrounding windows fill with their appropriate instantiated parts from the template, LTC Cogman is in a way thankful that this is an early version of AIE. From the product description material, he recalls that more windows are currently under development for the next version that will support the process all the way through to screen design. That would have meant even more things getting instantiated from his single command to create an instance of "make pressure". LTC Cogman is not sure that he would not have been overwhelmed by all that automation, given this was his first time using AIE. On the other hand, he is disappointed that he'll need to do that step by hand for now, and even more disappointed to think that he'll have to manage any Decision Centered Testing cases and results by hand as well.

B.3.4 Version Managing

Just to be on the safe side, LTC Cogman decides the 'save early/save often' mantra would be wise. As he selects the "Save Project" from the menu, the system prompts him for all the standard Configuration Management information that software engineers are now accustomed to: Version Number, Change Comments, etc. In this way, AIE is able to keep a full audit trail of each incremental change. This feature also enables LTC Cogman the ability to recover earlier versions for whatever reason.

B.4 Production Features

After completing all the specifics for the "Psychological Pressure" design thread (from FAN through CWRs to IRRs), LTC Cogman runs the Project Validation/Checking tools to ensure he hasn't missed any default items that need to be completed, hasn't left anything unlinked from its corresponding siblings in

the model, etc. (He was slightly embarrassed to see the system status window display the fact he'd forgotten to link "Psychological Pressure" into the original FAN structure, it was located in the right geographical area on the diagram, immediately below where he'd intended to link it, but it took the Completeness Checker to find that he'd not inserted the link itself.

Adding the link created a bit more CWR/IRR work in the design thread, and then another save. Now he was ready to get another set of eyes to be sure he was on a reasonable track. He selects "Output CSE Analysis Report (CSEAR)". AIE begins post-processing the model into the structure of the standard CSEAR report, basically extracting/converting the design database into a report, complete with figures, and designer notes.

With that report in hand, LTC Cogman leaves to find his SME and get a 'gut check' on how this new function has been modeled. Now if he just had that next version, he would have been able to insert a couple of display design concepts, much easier to show an SME than the ACWA methodology's first report. That next version will also output some initial software design report, like the Software Requirements Specification (SRS), as well as link directly to some basic software objects for the display system.

C AIE System Description

C.1 Current Project Scope Limit/Objectives

The IDE for cognitive engineers described in this document is not intended to be a complete design. Time and funding constraints has forced the design to focus on supporting the functions and cognitive work that occurs during the analysis of the domain and the design of the system. The cognitive engineering functions and cognitive work associated with evaluation of the developed decision support system has been left for a future design effort.

Even for the analysis and design functions, much work still needs to be done. The AIE system described in this document is an attempt to apply the "lessons learned" from our analysis of CACSE and other earlier work, and apply it to the development of an IDE for cognitive engineers. While this document is only a first step, in the development of AIE, it is important in that it provides the framework for future software development.

C.2 Envisioned System Overview

AIE will support cognitive engineers in capturing and maintaining the essential cognitive issues and relationships developed through an ACWA™ approach. It will support a robust CSE methodology adapted from Rasmussen (1986), yet will be sufficiently flexible to incorporate results from multiple, complementary cognitive analytic approaches into the IDE. This is designed to increase the maturity level of current approaches to Cognitive Systems Engineering and Cognitive Task/Work Analysis.

Moreover, AIE will be a tool for software developers to maintain awareness of the "design basis" underlying the resulting system requirements and specifications, by forming a maintainable, traceable component of the functional design. Thus, ***our vision is to develop an IDE that simultaneously and seamlessly aids the experienced CSE analyst in the modeling and documentation aspects of the ACWA™ process, and the equally experienced software developer, during the construction of the resulting DSS.***

Like other IDEs, AIE files will be organized around the concept of a project. While AIE may require multiple file types to store the information and objects entered by the cognitive systems engineer, all will be assessed via the project name. This IDE managed organization helps to minimize the difficulty of keeping the components of the ACWA™ process coherent and synchronized, as is the problem with current manual processes.

The primary benefit of an integrated, tool-supported process is expected to be the radical advance in the impact of ACWA™ outcomes on the resulting DSS design. The resulting system design, from user interface presentation to underlying representations, all the way to sensor placement will be based on the CSE analysis. To facilitate this, the development environment will be seamless, affordable to maintain, and dramatically improve a spiral, iterative design process. The envision AIE system also will result in tremendous gains in productivity across the ACWA™ methodology. This productivity improvement will result from both the automatic generation of the linked ACWA™ objects within an IDE, as well as the library of cases contained with the AIE system that the cognitive engineer can draw on when developing a new DSS.

C.2.1 Developing an Integrated Development Environment (IDE) for ACWA™

One of the factors that can affect the quality and cost of the application development significantly is how it is designed and developed. In order to improve the productivity of the developers and increase the likelihood of the system's success, software developers have turned to Integrated Development Environments (IDE) as a means to both manage complex projects and reduce their development time.

A software engineer can develop any application by purely writing text-based code without using any IDE. However, IDE's offer convenience and automation of code generating, resulting in much higher productivity. A good IDE should never offer poorer productivity for a software developer than using simple text editors to generate all necessary code. This also should be true for an IDE developed for a cognitive systems engineer.

In general, a good IDE should have the following characteristics:

- **Cross component dependencies.** Every time a developer accesses an element in an IDE, it should automatically update all related elements. The IDE should contain a knowledge base that enables it to know the impact of any changes.
- **High flexibility.** IDE's automate many business processes, but good IDE's should never disable the access to any platform feature that can be accessed through other means. In other words, IDE's should only empower, not weaken, the developers.
- **Rich set of features.** It is important that an IDE contain the tools necessary for system development. However, while it is easy to add features, it is much harder to add features without posing any threat to the system's robustness or increasing the developer's workload.
- **Powerful help tool.** Most IDE's contain a large set of features that few people can learn by heart. An excellent help tool can boost productivity significantly.

C.2.2 Integrated Development Environment Defined Roles

TogetherSoft, a prototypic IDE, uses roles to govern the access and interaction with the files in a software project. User role defines what features of the system are easily accessible. If users wanted access to features outside of their role, they can either change their roles or define a new role that has access to the features that they want. Features reside on panes or tabs; thus, the system limits access by selectively hiding these panes and tabs. TogetherSoft defines four roles for users of their software development environment:

- **Business Modeler** – whose role is intended for domain experts and analysts.
- **Designer** – whose role is both analysis of the domain and system design (both software and GUI). The user in this role has access to all features up to the point of writing software code.
- **Developer** – whose role is involved in all phases of application development, those who are analysts, designers, and programmers. In this role, the user interface provides ready access to all operations.
- **Programmer** – whose role is intended for those who are engaged only in programming. In the Programmer role, the system opens with a closed Designer pane. It is still possible to open the Designer pane if required.

One approach to developing an IDE for cognitive systems engineers would be to use roles to define the necessary system functions. However, rather than use of roles to define how users interact with the system, roles, in this case, are used to identify the types of tasks that a user in that role would need to

accomplish as well as the products that they need to produce. (See Figure 1, following.) The following five roles are to be supported within the IDE for cognitive engineers:

- **Analyst** – whose role is to interact with subject matter experts in the target domain, in order to collect and organize the knowledge necessary to construct the latter ACWA™ artifacts. The initial focus of users in this role is to compile the Knowledge Elicitation Domain Report (KEDR). The Analyst role also has to scrutinize the contents of the knowledge elicitation collection effort and use it to develop the three main artifacts of the ACWA™ analysis effort; the FAN, CWRs and IRRs. Development of these artifacts will lead directly to the development of the Cognitive Systems Engineering Analysis Report (CSEAR).
- **Designer** – whose role is to take the results of the Analyst and transform them into a design that supports cognitive work. The goal of this user is to transition the completed analysis and develop a set of "plans" that will enable a decision-aiding tool to be built. This role is responsible for the construction of the System Requirements Specification (SyRS), and Cognitive System Engineering Design Specification (CSEDS).
- **Bridger** – whose role it is to examine the documents produced by the Designer role, in order to transform them into software requirements. The elements created/captured in the proposed IDE for cognitive engineers to this point have been in developed in the ACWA framework. The goal of this user is to transition these cognitive elements into a software framework. This transition is captured in the initial version of the Software Requirements Specification (SRS) document.
- **Evaluator** – whose role it is to develop a series of tests to determine the impact of the final design on user-system decision-making performance. The goal of this user is to develop a scenario comprised of a series of cognitively demanding decision events and an experimental plan that to guide data collection and analysis. This role is responsible for the development of the Decision Centered Test Report (DCTR), which includes both the design of the evaluation program, as well as the results of the evaluation.
- **Reviewer** – whose role is to examine all aspects of the ACWA™ process and ensure that the artifacts that capture that process meet those specified by the methodology. This role is necessary because no formal metrics exist to computationally assess their quality. The goal of this user is to serve the same function as quality assurance metrics do in software analysis. This management function requires that the user be able to view all elements for the design of a system. This includes all inputs from the Analyst role through the Evaluator role.

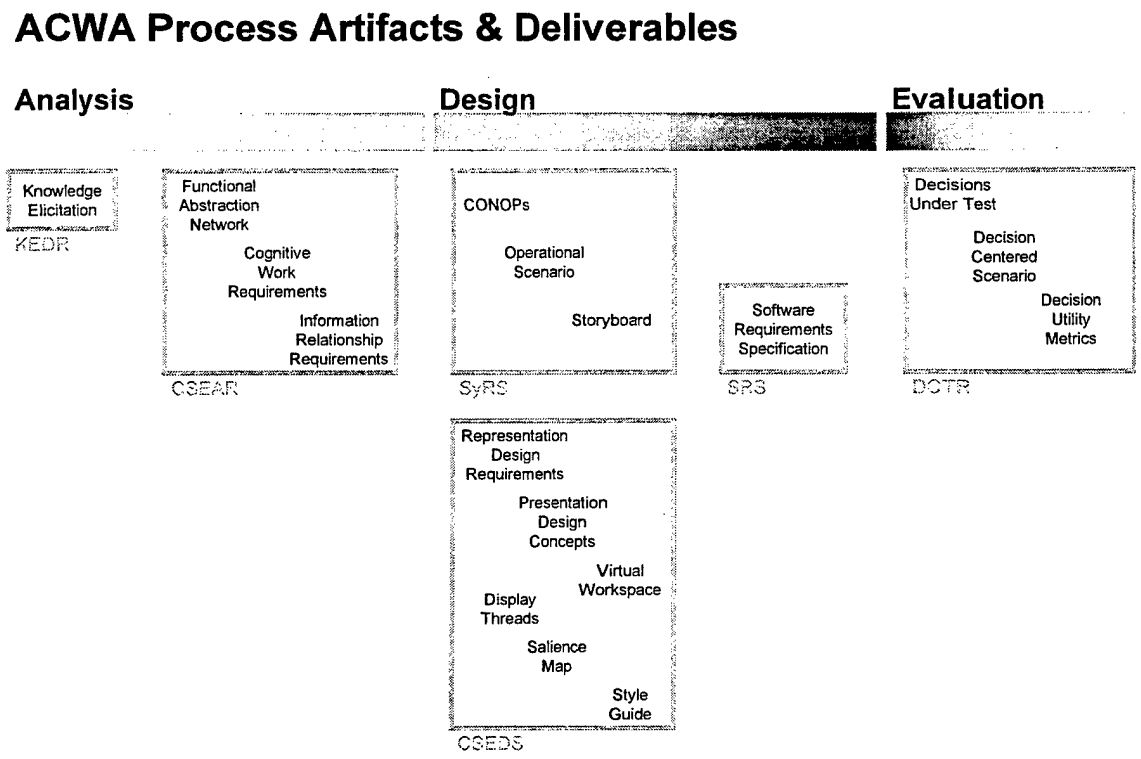

Figure 1. ACWA™ Process Artifacts and Deliverables

C.3 Envisioned Physical Work Space

Typically the design of systems requires that the developer take into consideration size limitations of the workspace. For software systems, this typically manifests itself in the minimum/maximum size of the display window. For this design effort, ***ManTech Aegis Research Corporation has chosen not to place any constraints on the size of the workspace.***

By designing using an unconstrained workspace, some of the serialism and parallelism issues are left for future design refinements. For the purposes of the AIE design effort, ManTech Aegis Research Corporation will assume that AIE will exist on single continuous pane of glass. Removing the workspace constraint allows for greater freedom in the system design, and provides ManTech Aegis Research Corporation the opportunity to explore many alternative configurations to the AIE system. ManTech Aegis Research Corporation realizes that this is an unrealistic assumption, but given the limited time and budget available for this design effort, it was a reasonable solution to help manage the scale and scope of this project.

C.4 Envisioned Virtual Work Space

As noted in the prior section, ManTech Aegis Research Corporation will design the envisioned virtual workspace under the assumption that there are no physical constraints on the display's size. Using this assumption, the AIE's workspace design is shown in Figure 2. Please note that the virtual workspace shown in Figure 2 does not contain heading and menu bars.

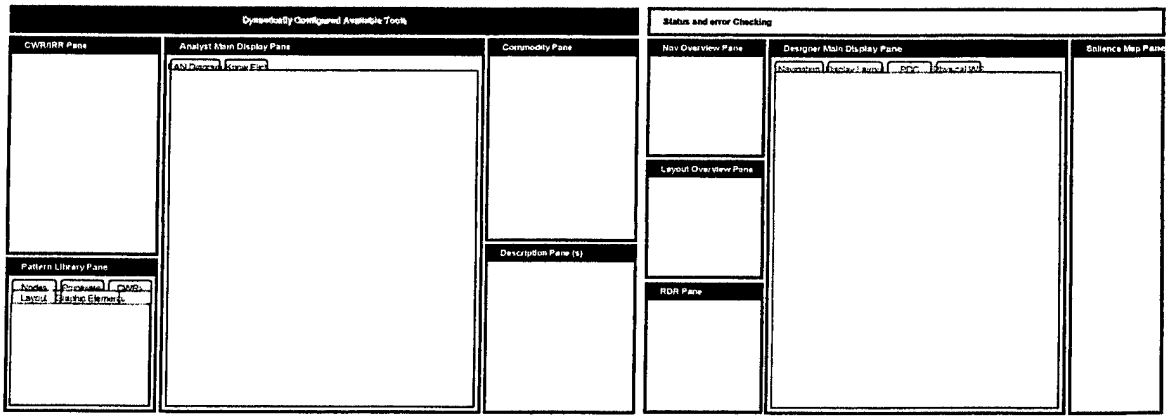

Figure 2. AIE Workspace Design

The workspace is composed of twelve (12) panes. As an IDE, each of the panes in the workspace design of AIE is potentially impacted by actions/modifications in any other pane. The impact of interactions in each of the panes will be discussed in the following section. The following paragraphs provide a brief description of the twelve panes working from top to bottom and left to right.

Along the top of the workspace are the **Tools Pane** and **Status and Error Checking Pane**. The **Tools Pane** displays the set of available tools, based on which of the other panes is currently active. Each pane has its own set of tools. For example a text-based pane, like the **CWR/IRR Pane,** will have one set of tools, while a diagramming-based pane, like the **Designer Main Display Pane,** will have another.

The **Status and Error Checking Pane** displays the currently selected object's name, as well as any available information about its state. This pane will indicate both approval status, as well as any errors in the selected object.

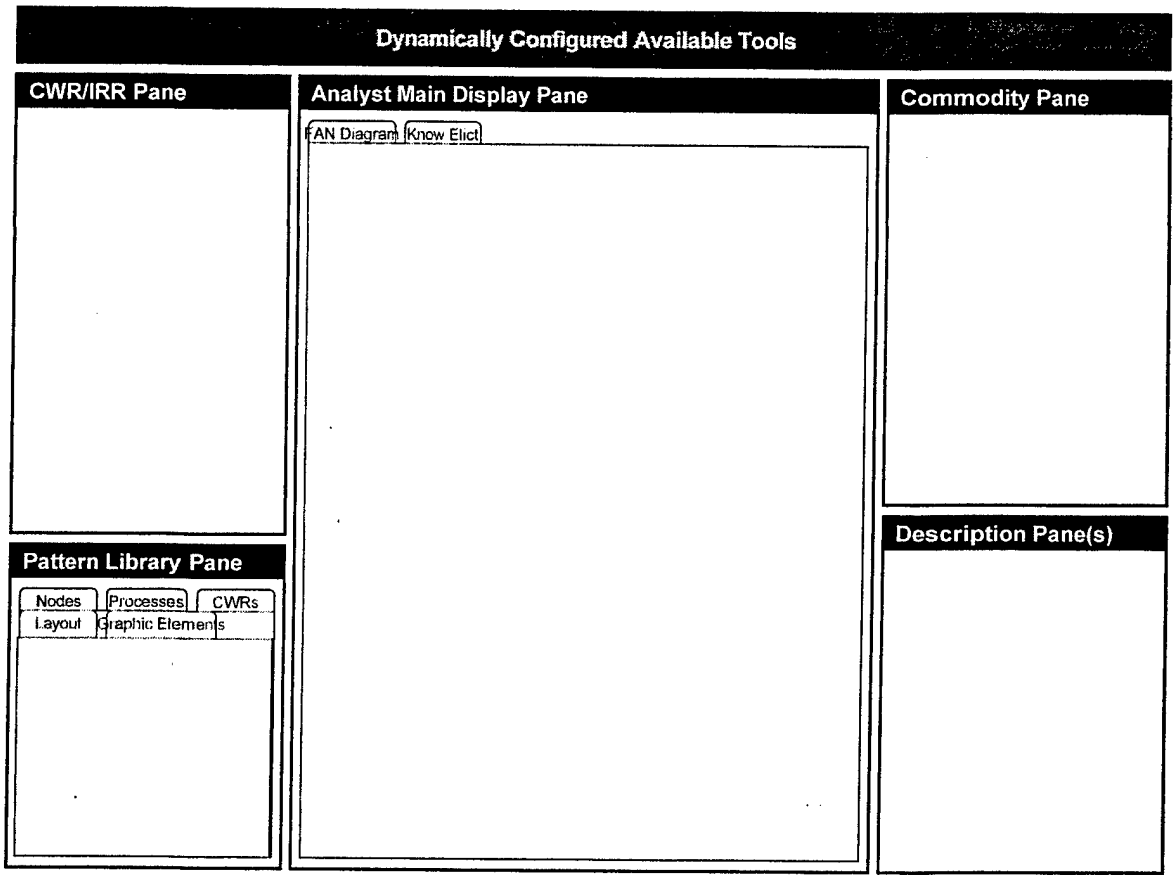

Figure 3. Left-Hand Side of Virtual Workspace

Along the left-hand side of the workspace directly below the **Tools Pane** is the **CWR/IRR Pane**. Refer to Figure 3. This pane portrays all of the CWR and IRR objects contained within the project, sequentially, in this scrollable pane.

Below the **CWR/IRR Pane** along the left-hand side is the **Pattern Library Pane**. The **Pattern Library Pane** is a tabbed pane containing five tabs, which provides the cognitive systems engineer access to various templates. These templates are meant to foster reuse and quicken the system development process. The initial AIE design envisions five pattern types:

- Goal-Process Node Templates,
- Process Model Templates,
- Cognitive Work Requirement Templates,
- Layout Templates, and
- Graphic Element Templates.

Moving to the left is the **Analyst Main Display Pane**. It also is a tabbed pane. The **Analyst Main Display Pane** is envisioned to have two tabs: (1) for managing the results from the project's knowledge elicitation effort, and (2) for developing and managing the project's Functional Abstraction Network (FAN).

Adjacent to the **Analyst Main Display Pane** is the **Commodity Pane**. This pane lists all of the commodities that have been specified in the FAN. This scrollable pane consists of two columns: (1) an alphabetical list of commodities, and (2) the corresponding definitions.

Below the **Commodity Pane** is the **Description Pane** (in the detailed description of AIE components, this component's name is changed to **Object Attribute Pane**, Sub-Section D.3.1). This pane provides descriptive information about the selected object. This object can be any element in any of the other panes of the AIE. What is portrayed in this pane differs, based on the type of object selected. For example, if a tool icon is selected from the **Tool Pane,** its name and use might be displayed in the **Description Pane.** If, on the other hand, a goal-process node is selected from within the **Analyst Main Display Pane**, then a goal description, along with the description for any support/supported links, may be provided. Alternately, some objects within the AIE system may portray an attribute value table.

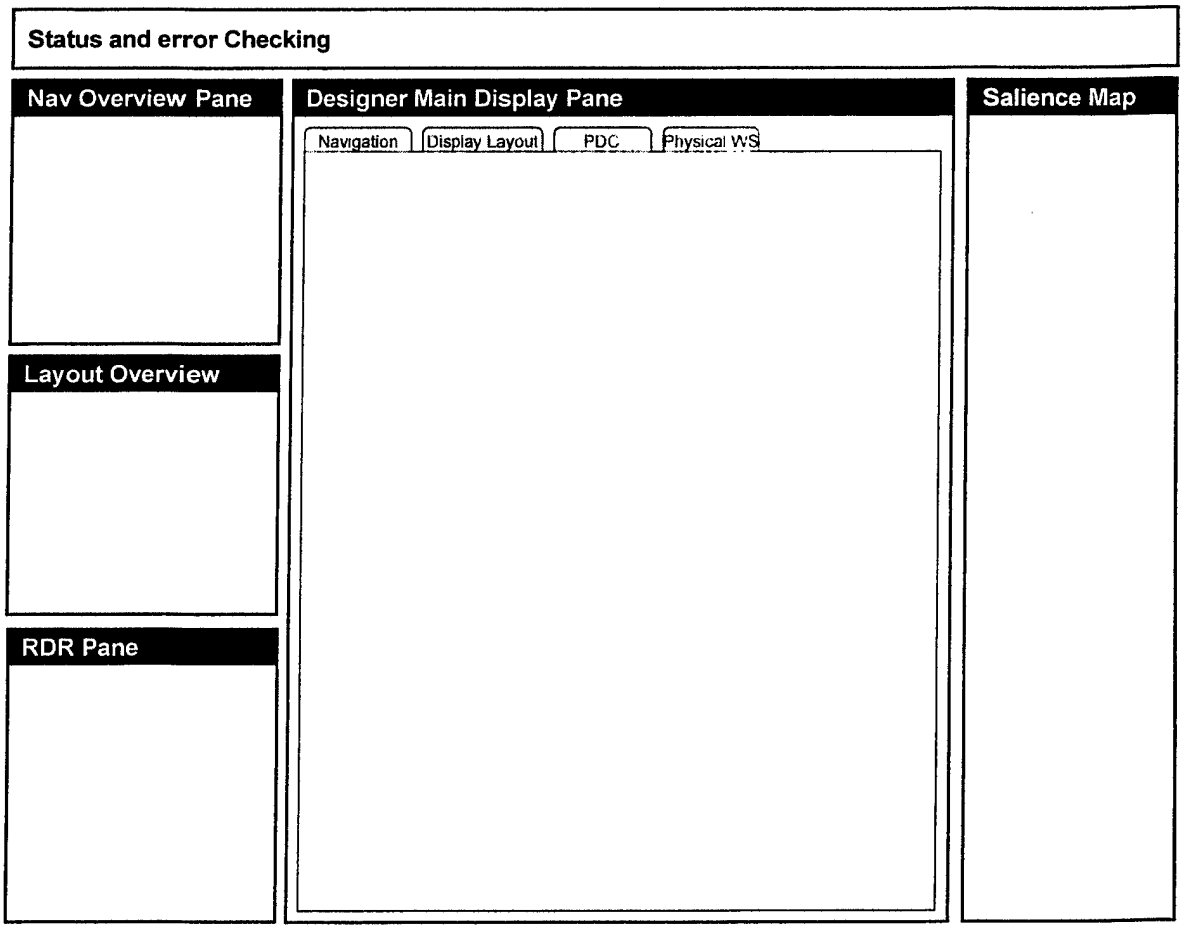

Figure 4. Right-Hand Side of Virtual Workspace

To the right of the **Commodity Pane** are the **Navigation Overview Pane** and the **Display Layout Overview Pane.** (Refer to Figure 4, above.) Both of these panes provide thumbnail views of their perspective topic areas. The **Navigation Overview Pane** displays the location of the view being portrayed in the **Designer Main Display Pane**. This view differs depending on the tab selected in the **Designer Main Display Pane**. The **Display Layout Overview Pane** behaves similarly, but is only active when the either the Display Layout or PDC tab is selected in the **Designer Main Display Pane.**

To the left of the **Designer Main Display Pane,** and directly below the **Display Layout Overview Pane,** is the **RDR** (Representation Design Requirements) **Pane**. This pane portrays all of the RDR objects contained within the project, listed sequentially, in this scrollable pane.

Moving further to the left is the **Designer Main Display Pane**. Like the **Analyst Main Display Pane**, this is a tabbed pane. The pane is envisioned to have four tabs, each providing the cognitive systems

engineer the ability to construct a different type of diagram. The four tabs envisioned in the initial AIE design are:

- Navigation Diagram (Workspace 1),
- Display Layout Design (Workspace 2),
- Presentation Design Concepts, and
- Physical Workspace Design Diagram,

The pane to the far-most right is the **Salience Map Pane**. This pane lists all of the graphic components contained within a display layout. This pane enables the cognitive systems engineer to record the salience values for these components.

The twelve panes listed in the preceding paragraphs contain an initial assessment of the features necessary for a cognitive systems engineer to accomplish the cognitive demands of their work domain. The individual components of this virtual workspace are described in greater detail in Section D. This initial virtual workspace design is not complete, and will likely be expanded on as development on AIE continues.

C.4.1 Display Structures

This section is not part of the current phase's work. It will be specified in future phases.

C.4.2 Display Real Estate Allocation

This section is not part of the current phase's work. It will be specified in future phases.

C.5 System Navigation Map

This section is not part of the current phase's work. It will be specified in future phases.

C.6 Style Guide

This section is not part of the current phase's work. It will be specified in future phases.

C.6.1 Font Chart

This section is not part of the current phase's work. It will be specified in future phases.

C.6.2 Standard Features

This section is not part of the current phase's work. It will be specified in future phases.

C.6.3 Standard Colors / Shapes / Fills

This section is not part of the current phase's work. It will be specified in future phases.

D INDIVIDUAL DISPLAY DESCRIPTIONS

The detailed descriptions of the individual panes that comprise AIE are organized into three subsections/categories. In the first subsection, the panes within AIE that relate primarily to *Analysis* of the work domain are discussed. In the second subsection, the panes that relate primarily to system *Design* are discussed. In the third subsection, the panes that serve management functions or are applicable to both *Analysis* and *Design* are discussed.

D.1 Analysis Panes

In the virtual workspace design, most of the *Analysis* panes are located on the left-hand side of the workspace. Figure 5, below, highlights the panes that are discussed in this section.

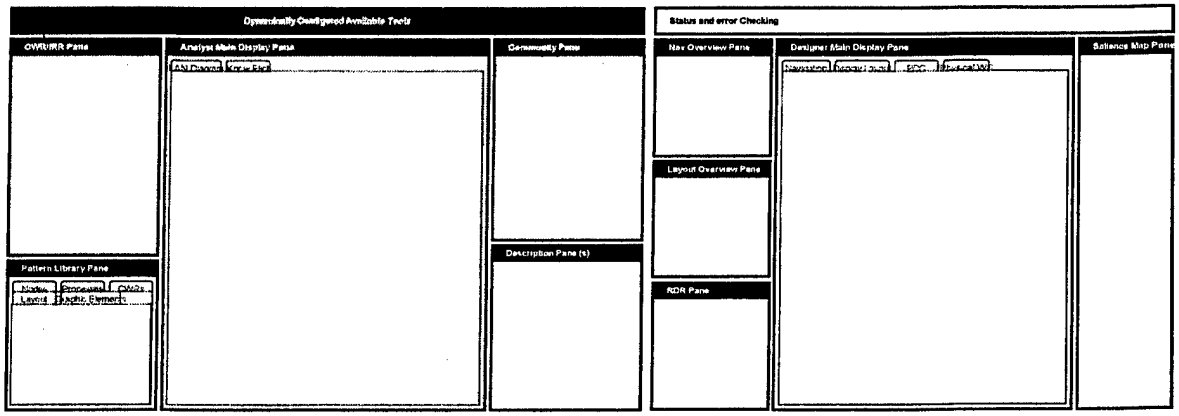

Figure 5. Analysis Panes Location Within AIE's Workspace Design

D.1.1 Analyst Main Display Pane

The **Analyst Main Display Pane** is a tabbed pane consisting of two tabs, a *Knowledge Elicitation* tab and a Functional Abstraction Network (*FAN*) tab. The *Knowledge Elicitation* tab is used primarily for the entry and management of text, while the *FAN* tab is primarily a diagramming tool.

D.1.1.1 Design Objective

The purpose of this pane is to provide the CSE analyst a large workspace to record information and develop the structure of the functional space.

The purpose of the *Knowledge Elicitation* tab is to provide a repository to record and organize insights gained/collected during the knowledge elicitation (KE) process, so that they may be utilized during "downstream" activities of the ACWA™ process. The *Knowledge Elicitation* tab will provide a storehouse for both the raw data collected during KE activities, as well as the consolidated analysis of significant KE insights derived from that raw data. These KE insights will become AIE objects that can then be associated with other AIE objects within the project. Figure 6 represents an example of a KE insight.

Identification is an Essential Support to Threat Evaluation

The process of determining the Identity (i.e., determination of the type and country) of an object needs to be conceptually separated from the evaluation of an object's threat level. While it is true that the identity of an object is an important input to the evaluation of the object's threat level, experts make use of the identity of an object in several other ways. Having some confidence in knowing the identity of an airborne object gives the expert more confidence in the evaluation of the object's threat level, in its weapons capability, and, therefore, in his/her selection of an appropriate response to it. The result is that the Identification process is important enough that it needs to be considered as a separate entity within this problem domain.

Figure 6. Example Knowledge Elicitation Object

The purpose of the *FAN* tab is to provide a diagramming environment that will permit the cognitive systems engineer to develop and edit a functional representation of the domain using the ACWA™ methodology. The *FAN* tab will present a network diagram of the problem domain that serves as a starting point for understanding fundamental relationships, scope of the problem space to be modeled, and essential objectives. This representation is composed of numerous linked AIE objects. Potential AIE objects include:

- Goal Objects,
- Commodity Objects,
- Processes Objects,
- Transport Objects, and
- Support/Supported Link Objects.

The relationship amongst these AIE objects can be seen in Figure 7, below. Some of these objects are nested within one another, while other AIE objects are used to connect one AIE object to another.

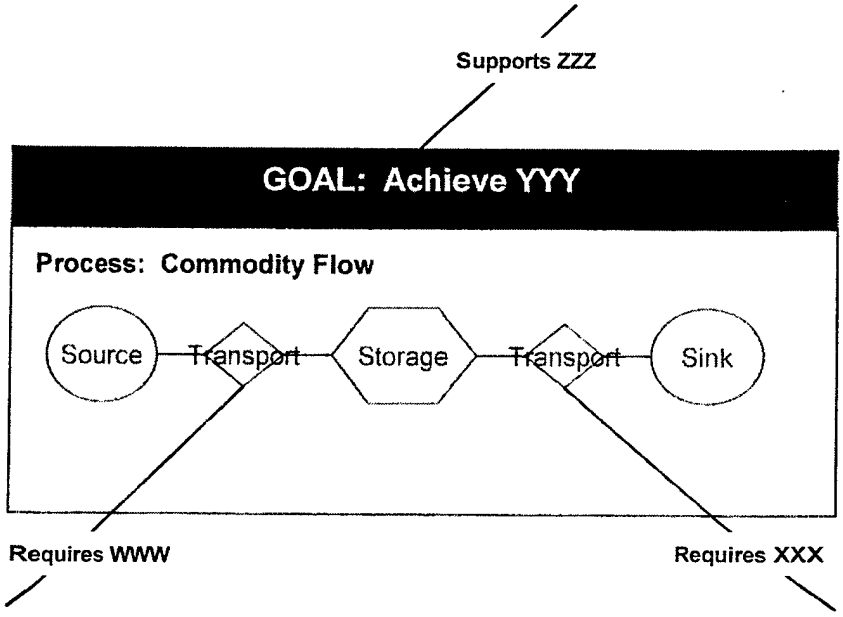

Figure 7. FAN node with Support / Supported Links

Together, these objects create a decomposition network of nodes that are linked together to form abstract goals to specific means in order to achieve these goals. Figure 8 provides and example of a FAN. This model provides the essential, decision-centered framework for the development of additional ACWA™ artifacts and associated AIE objects.

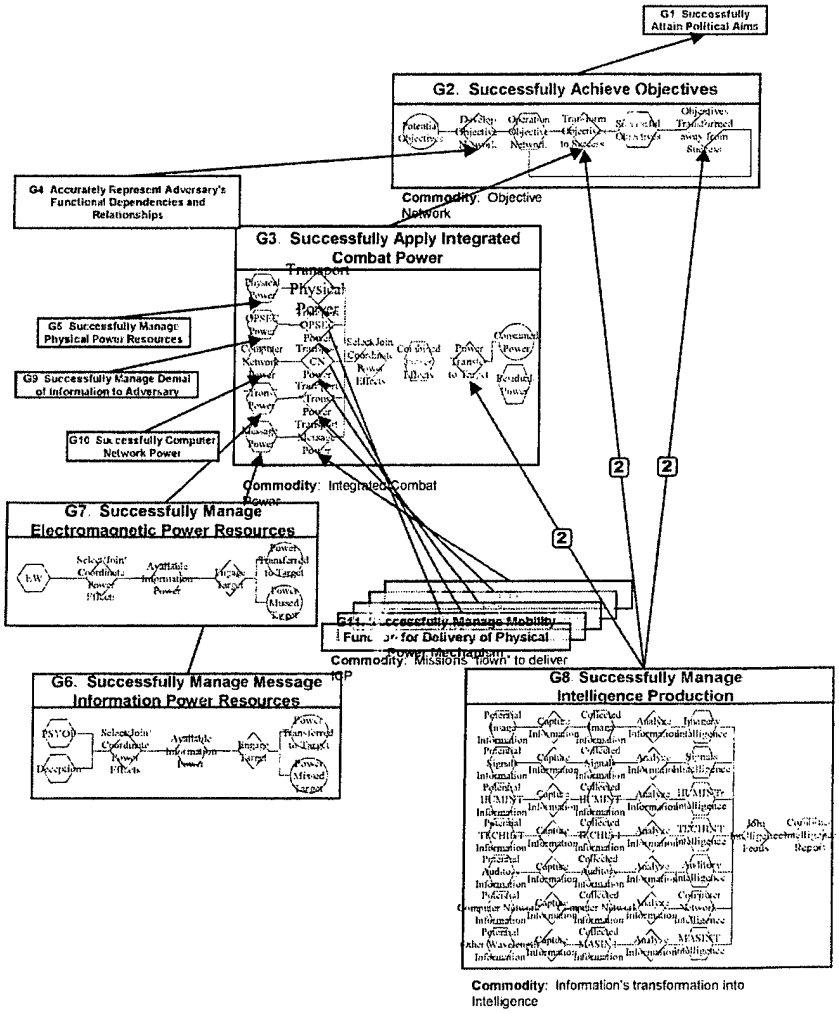

Figure 8. Example Functional Abstraction Network

D.1.1.2 Virtual Workspace Design

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.2.1 Window Attributes

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.2.2 Menu Bar

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.2.3 Navigation Controls

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.2.4 Display Controls

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.2.5 Display States

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.2.6 Pop-up Messages or Pop-up Displays

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.3 Representation Design Requirements

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.3.1 Information Relationship Requirements

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4 Presentation Design Concept (PDC) Description

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4.1 Screen Design

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4.2 Salience Map

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4.3 Display Elements

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4.3.1 Tab Sequence

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4.4 Data Description/ Requirements

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.4.5 Scale/Scope

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.5 Interaction Needs

D.1.1.5.1 Operator Inputs

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.1.5.2 Knowledge Elicitation Tab

The primary input for cognitive engineer in this tab will be text. Selecting the tab will bring up a scrollable text field. The tools available for use via the **Tool Pane** will be similar to those used in most other simple text editors. In addition, the *Knowledge Elicitation* tab will support the designation of portions of text as a KE object. The cognitive engineer will be able to link this AIE object to other AIE objects within the project.

D.1.1.5.2.1 Local Automation

The *Knowledge Elicitation* tab provides only limited automatic services for the cognitive engineer. Text that is designated as a KE insight object will be automatically numbered, based on its order of designation. Once a portion of the text has been designated as a KE insight object, how that text is displayed will be modified to reflect its change from a simple text string to a relevant AIE object.

D.1.1.5.2.2 Interaction with Other Panes

KE insight objects that have been associated with other AIE objects will be highlighted when one of those is selected. For example if a KE insight object has been associated with a particular cognitive work requirement (CWR), selecting that CWR will cause the text in the *Knowledge Elicitation* tab to auto scroll to that portion of the text field that contains the KE insight. If the *Knowledge Elicitation* tab is not active when a linked AIE object is selected, a message will appear in the **Status and Error Checking Pane** to inform the cognitive engineer of the link.

The designation of a KE insight object will not automatically cause the creation of any other AIE objects.

D.1.1.5.3 Functional Abstraction Network (FAN) Tab

The *FAN* tab allows the cognitive engineer to assemble diagrams quickly using predefined object shapes. Selecting this tab will bring up a scrollable view port that will permit the construction of the project's functional model. The tools available for use via the **Tool Pane** will contain each of the type of objects that need to be created to construct a FAN. The *FAN* tab will permit the CSE analyst to create a series of AIE objects, and relate them to one another, in order to develop a functional skeleton onto which other AIE objects associated with the project will be linked.

Each of the following actions will generate an AIE object. These objects will interact with one another within the *FAN* tab, as well as with AIE objects in other panes. Within the *FAN* tab of the **Analyst Main Display Pane,** the CSE analyst will be able to:

- Create a Goal-Process Node
- Specify the Goal Statement for the Goal-Process Node
- Specify the Commodity that is the Focus of a Goal Process Node
- Specify the Flow Model for the Process of a Goal-Process Node

- Add Source node(s) for Flow Model
- Add Transport node(s) to the Flow Model
- Add Storage node(s) to the Flow Model
- Add Sink node(s) to the Flow Model
- Add Links between the Process nodes to create the Flow Model
- Specify Links between the Goal-Process Nodes
- Create Explanative Paragraph for Goal-Process Node
- Create Explanative Paragraph for Links between Goal-Process nodes
- Associate orphan FAN objects with previously established FAN objects

D.1.1.5.3.1 Local Automation

The *FAN* tab provides some automatic services for the cognitive engineer. The creation of some AIE objects within the *FAN* tab will lead to the creation of other objects within the *FAN* tab. The AIE system will provide the following automated support within the *FAN* tab of the **Analyst Main Display Pane**:

- Number Goal-Process nodes sequentially based on order of creation
- Create default Goal Statement on Goal-Process node creation
- Create default Commodity on Goal-Process node creation
- Number Process nodes based on position in Flow Model and Goal-Process node
- Create default name for Process nodes based on type and Commodity name
- Alert CSE analyst for the need to enter a Goal-Process explanation paragraph on Goal-Process Node creation
- Alert CSE analyst for the need to enter Link explanation paragraphs (from both support and supported ends) on Link creation
- Create AIE object association between Flow Model nodes and Goal-Process nodes
- Highlight local impact of selection of associated AIE objects made in other panes

D.1.1.5.3.2 Interaction with Other Panes

The creation of AIE objects within the *FAN* tab automatically leads to the creation of AIE objects in other panes. The following is a partial list of the automatic features that the AIE system will provide when AIE objects are manipulated within the *FAN* tab of the **Analyst Main Display Pane**:

- Create default CWR & IRR objects on the creation of a Goal-Process Node
- Create default CWR & IRR objects on the creation of a Link between Goal-Process Nodes
- Create default CWR & IRR objects on the creation of Flow Model nodes
- Modify CWR & IRR objects based on changes made to Commodity object
- Create default PDC object on the creation of a Goal-Process Node
- Create default Display Layout object on the creation of a Goal-Process Node
- Create default Navigation object on the creation of a Goal-Process Node
- Create entry in Commodity Dictionary on the creation of a Goal-Process Node

Actions within other AIE panes will impact AIE objects within the *FAN* tab of the **Analyst Main Display Pane.** Again, this is only a partial list of the features that the AIE system will provide.

- Create a Goal-Process node for orphan CWR objects

- Instantiate Template imported from Pattern Library with appropriate FAN objects
- Create a Goal-Process node for orphan PDC objects
- Create a Goal-Process node for orphan Commodity objects

D.1.1.6 Dynamic Description

This section is not part of the current phase's work. It will be specified in future phases. This section will be excluded from the remaining display descriptions.

D.1.2 CWR/IRR Pane

The **CWR/IRR Pane** is a scrolling pane, which contains a tree-like component for displaying cognitive work requirements (CWR) and information resource requirements (IRR). The **CWR/IRR Pane** is used primarily for the entry, display, and management of text. However, each CWR also will have an attribute coding to indicate its decision type.

D.1.2.1 Design Objective

The purpose of this pane is to provide the CSE analyst a workspace to simultaneously present CWR and IRR while developing and modifying the FAN. The **CWR/IRR Pane** is a repository to develop, modify, and organize the cognitive work that needs to be accomplished related to a Goal-Process node. This cognitive work becomes a CWR object. This AIE object then can be associated with other AIE objects within the project. This pane also will allow the CSE analysts to specify the complete set of information necessary to complete the cognitive work within the CWR. Figure 9 provides an example of CWRs and IRRs.

CWR2_P5.9	Integrate the Indicator assessments into a single Objective assessment	
	IRR2_P5.9_1	Individual indicators associated with Objective
	IRR2_P5.9_2	Algorithmic rule for combining indicators
CWR2_P5.10	Assess Indicator state	
	IRR2_P5.10_1	Evidence contained in Intelligence Report conditionally based on THREATCON level
	IRR2_P5.10_2	Evidence contained in Intelligence Report conditionally based on INFOCON level
	IRR2_P5.10_3	Evidence contained in Intelligence Report conditionally based on FPCON level
CWR2_P5.11	Evaluate the need to override "algorithm" used to integrate the Indicator assessments	
	IRR2_P5.11_1	Algorithmic output of objective state compared to analyst's expectation for objective state
	IRR2_P5.11_2	Relative weighting of indicators within algorithmic rule for combining indicators

Figure 9. Example Cognitive Work Requirements and Information Relationship Requirements

The **CWR/IRR Pane** also will support the ability to categorize the CWR by type. The purpose of this feature is to provide the CSE analyst an audit of the types CWR utilized and to ensure that all appropriate CWR types have been considered. Figure 10 provides an example of CWR decision type table.

CWR#	CWR Details	Goal Monitoring	Process Monitoring	Automation Supervision	Process Control	Manual Takeover	Abnormality Detection
CWR3_G.1	Monitor the availability of Integrated Combat Power to achieve **Objectives** (*Not Supported*)	●					
CWR3_G.2	Monitor composition of Integrated Combat Power (IW type) used for achieving **Objectives** (*TED & RON*)	●					
CWR3_G.3	Monitor Integrated Combat Power inventory level (*Not Supported*)	●					
CWR3_P6.1	Select/Coordinate available Combat Power to obtain an Integrated Combat Power level sufficient to cause the **Effects** necessary to support achieving **Objectives** (*Not Supported*)				●		
CWR3_P6.2	Monitor the synchronization between application(s) of Integrated Combat Power (*TED*)		●				
CWR3_P6.3	Monitor the synchronization between application of Integrated Combat Power and the **Effects** from prior **Mission-Actions** (*TED*)		●				
CWR3_P7.1	Monitor the mapping of Integrated Combat Power (through the mechanism of **Mission-Actions**) to **Targets** (*TED*)		●				
CWR3_P7.2	Monitor flow of Integrated Combat Power level (*Not Supported*)		●				
CWR3_P8.1	Develop projection of likely **Effects** resulting from the application of Integrated Combat Power (*TED*)				●		
CWR3_P8.2	Monitor the transfer of Integrated Combat Power (through the mechanism of **Mission-Actions**) to the **Target**, resulting in **Effect(s)** (*Not Supported*)		●				
CWR3_P8.3	Monitor the temporal pacing of the transfer of Integrated Combat Power (**Mission-Actions**) to the **Target** (*TED*)		●				
CWR3_P9.1	Monitor the temporal pacing of **Effects** resulting from the application of Integrated Combat Power (*Not Supported*)		●				

Figure 10. Example of CWR Decision Type Audit

D.1.2.2 Interaction Needs

The primary input for cognitive engineer in the **CWR/IRR Pane** will be text. CWR and IRR objects either can be entered directly by the CSE analyst, or they can be auto generated by the input of other AIE objects. The tools available for use via the **Tool Pane** will include those necessary for simple text editing functions, as well as tools necessary for creating CWR and IRR objects. In addition, the **CWR/IRR Pane** will support the designation of a CWR object as a particular decision type. The cognitive engineer will be able to link this AIE object to other AIE objects within the project. The **CWR/IRR Pane** also will support the expansion and contraction of CWR and IRR objects within the tree.

The **CWR/IRR Pane** will enable the CSE analyst the ability to specify Defined Concepts. Defined Concepts are text strings that have special meaning within the ACWA™ process, or have been defined as special terms within the within the particular project (e.g., Commodity). The AIE system will display Defined Concepts differently than other text to reflect its status as a relevant AIE object

This pane also will provide the ability to specify associations between the CWR and IRR objects to other AIE objects within the project. The following is a partial list of the actions the **CWR/IRR Pane** will enable:

- Create AIE object association between CWR objects and IRR objects
- Create AIE object association between CWR objects and FAN objects
- Create AIE object association between CWR objects and RDR objects

- Create AIE object association between CWR objects and PDC objects

D.1.2.2.1 Local Automation

The **CWR/IRR Pane** provides some automatic services for the cognitive engineer. The creation or modification of CWR objects within the **CWR/IRR Pane** will lead to the creation of IRR objects within the **CWR/IRR Pane**. The AIE system will provide the following automated support within the **CWR/IRR Pane**:

- Number CWR and IRR objects sequentially based on their association with FAN objects
- Create default IRR object based on CWR object creation
- Highlight Commodities and CSE Reserved Words in their CWRs
- Highlight local impact of selection of associated AIE objects made in other panes

D.1.2.2.2 Interaction with Other Panes

The creation of AIE objects in the **CWR/IRR Pane** automatically leads to the creation of AIE objects in other panes. The following is a partial list of the automatic features that the AIE system will provide when AIE objects are manipulated within the **CWR/IRR Pane**:

- Create default Goal-Process Node on the creation of orphan CWR & IRR objects
- Modify FAN objects based on changes made to Commodity object
- Create default PDC object on the creation of CWR & IRR objects
- Create default Display Layout object on the creation of CWR & IRR objects
- Create default Navigation object on the creation of CWR & IRR objects
- Create entry in Commodity Dictionary on the modification creation of a unique CWR object

Actions within other AIE panes will impact AIE objects within the **CWR/IRR Pane.** Again, this is only a partial list of the features that the AIE system will provide.

- Create default CWR & IRR objects on the creation of a Goal-Process Node
- Create default CWR & IRR objects on the creation of a Link between Goal-Process Nodes
- Create default CWR & IRR objects on the creation of Flow Model nodes
- Modify CWR & IRR objects based on changes made to Commodity object
- Instantiate Template imported from Pattern Library with appropriate CWR & IRR objects
- Create CWR & IRR objects for orphan PDC objects
- Create CWR & IRR objects for orphan Commodity objects

D.1.3 Commodity Dictionary Pane

The **Commodity Dictionary Pane** is a scrollable pane with two linked columns. The first column is an alphabetical list of all of the commodities in the project opened in the AIE system. The second column contains the definitions for the commodities listed in the first column.

D.1.3.1 Design Objective

The purpose of the **Commodity Dictionary Pane** is to provide a repository to record commodities contained with the open project, as well as to ensure that each Goal-Process Node with the project FAN contains a unique and fully specified commodity. This will provide an unambiguous understanding of each Commodity in the project, and will avoid multiple Goal-Process Nodes with same Commodity.

D.1.3.2 Interaction Needs

The primary input for CSE analyst in the **Commodity Dictionary Pane** will be text. The tools available for use via the **Tool Pane** will be similar to those used in most other simple text editors. The CSE analyst will be able to associate the Commodity object with other AIE objects within the project. Primarily, the CSE analyst will associate the Commodity object with a Goal Process Node object.

D.1.3.2.1 Local Automation

The **Commodity Dictionary Pane** provides only limited automatic services for the cognitive engineer. Definitions that contained references to other Commodities will display the Commodity object different than the surrounding text.

D.1.3.2.2 Interaction with Other Panes

Selecting **Commodity Dictionary Pane** entries will cause the highlighting of associated AIE objects. Similarly, if a Commodity object is modified in the **Commodity Dictionary Pane,** all instances of that Commodity within the project will be modified. The following is a partial list of the automatic features that the AIE system will provide, when AIE objects are manipulated within the **Commodity Dictionary Pane**:

- Create default Goal-Process Node on the creation of new Commodity
- Modify FAN objects based on changes made to Commodity object

D.2 Design Panes

In the virtual workspace design, most of the **Design** panes are located on the right-hand side of the workspace. Figure 11, below, highlights the panes that will be discussed in this section.

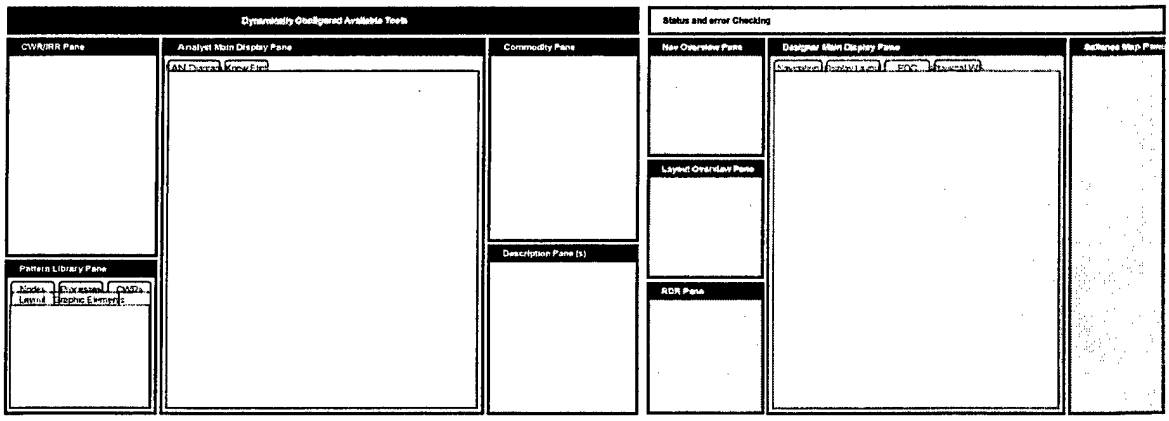

Figure 11. Design Panes Location Within AIE's Workspace Design

D.2.1 Designer Main Display Pane

The **Designer Main Display Pane** is a tabbed pane consisting of four tabs: (1) *Navigation Diagram* (Workspace 1) (2) *Display Layout Design* (Workspace 2) (3) *Presentation Design Concepts* and (4) *Physical Workspace Design.* Each of the tabs in the **Designer Main Display Pane** is focused on constructing a different type of diagram for the development of the decision aiding system that is the focus of the project.

D.2.1.1 Design Objective

The purpose of this pane is to provide the CSE analyst with a large workspace to develop the relationship between the components that will comprise the final decision aiding system. Each of the tabs in the **Designer Main Display Pane** provides a different level of granularity for this development.

The purposes of the *Presentation Design Concepts* (PDC) tab is to provide a diagramming environment that will permit the CSE designer the ability to develop and edit graphic elements for use in the project's decision aiding system, and associate those graphic elements with representation design requirement (RDR) objects, CWR objects and IRR objects. The *Presentation Design Concepts* tab will provide a storehouse for both the unstructured PDC objects, as well as the PDC objects that have been aggregated for inclusion in a single display or pop-up window. The *Presentation Design Concepts* tab also will enable the CSE designer to combine PDC objects, as well as import graphics from other programs. Figure 12 represents an example of several graphic forms combined into a single PDC.

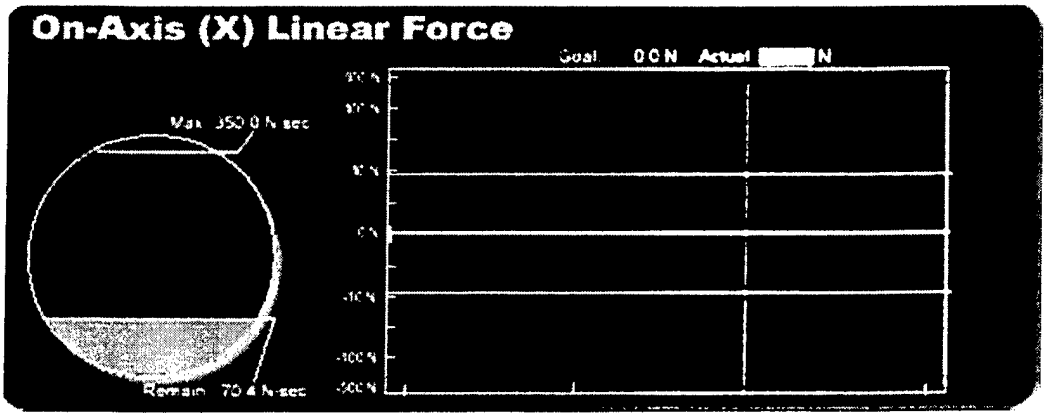

Figure 12. Example Presentation Design Concept

The purpose of the *Display Layout Design* tab is to provide a diagramming environment that will permit the CSE designer the ability to take the PDC objects that have been aggregated together in the *Presentation Design Concepts* tab, and develop and edit a spatial representation of how they will appear in one of the decision aid's windows. The window objects developed in the *Display Layout Design* tab serve as a starting point for developing the decision aiding system's navigation map in the *Navigation Diagram* tab. Figure 13 represents an example of a Display Design Layout.

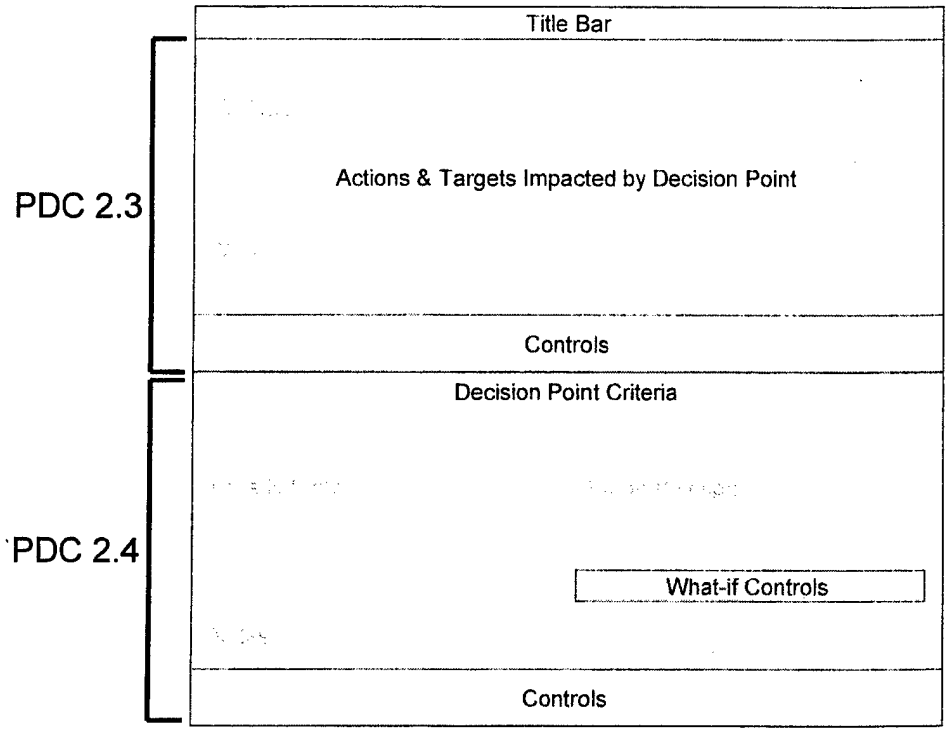

Figure 13. Example Display Design Layout (Workspace 2)

The purpose of the *Navigation Diagram* tab is to provide a diagramming environment that will permit the CSE designer the ability to take the window objects that have been created in the *Display Layout Design* tab and develop a navigation scheme of how users of the decision aiding system move between them. A second purpose of the *Navigation Diagram* tab is to provide the cognitive system engineer the ability to check the functional coverage of the project's decision aiding system design. Figure 14 represents an example of a Navigation Map.

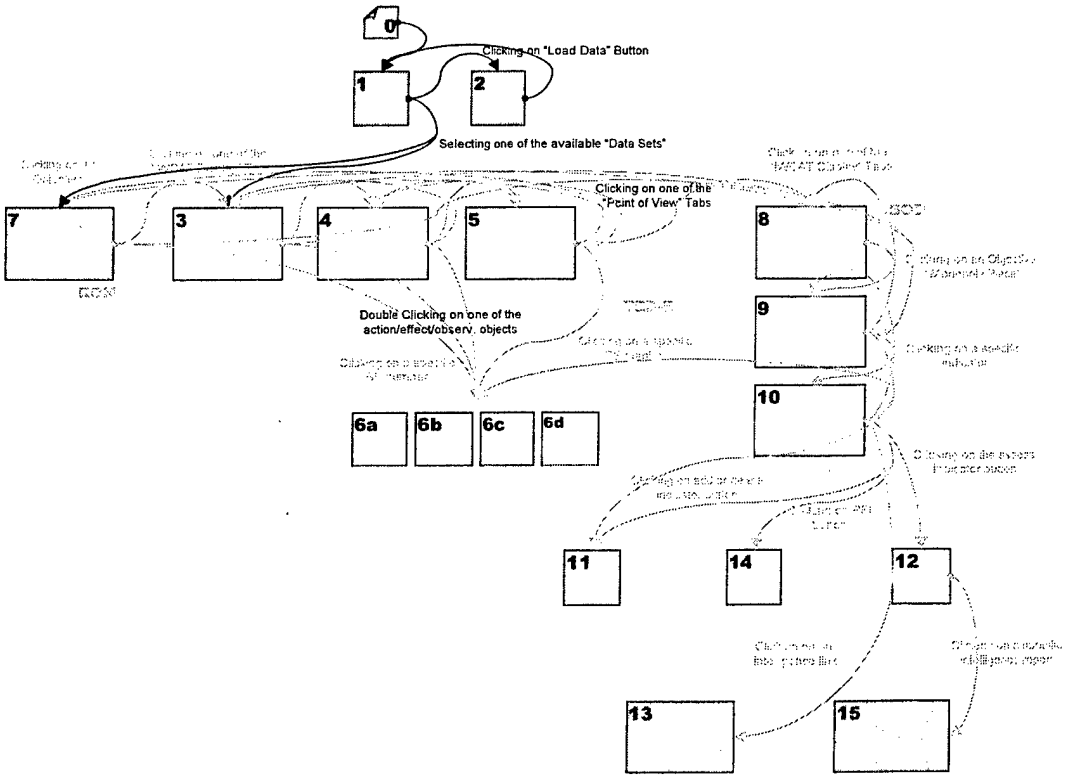

Figure 14. Example of a Decision Aiding System Navigation Map

The purpose of the *Physical Workspace Design* tab is to provide a diagramming to environment to illustrate how the decision aiding system developed for the project will be incorporated into the current or envisioned physical environment. Figure 15 represents an example of a physical workspace design developed using currently available tools.

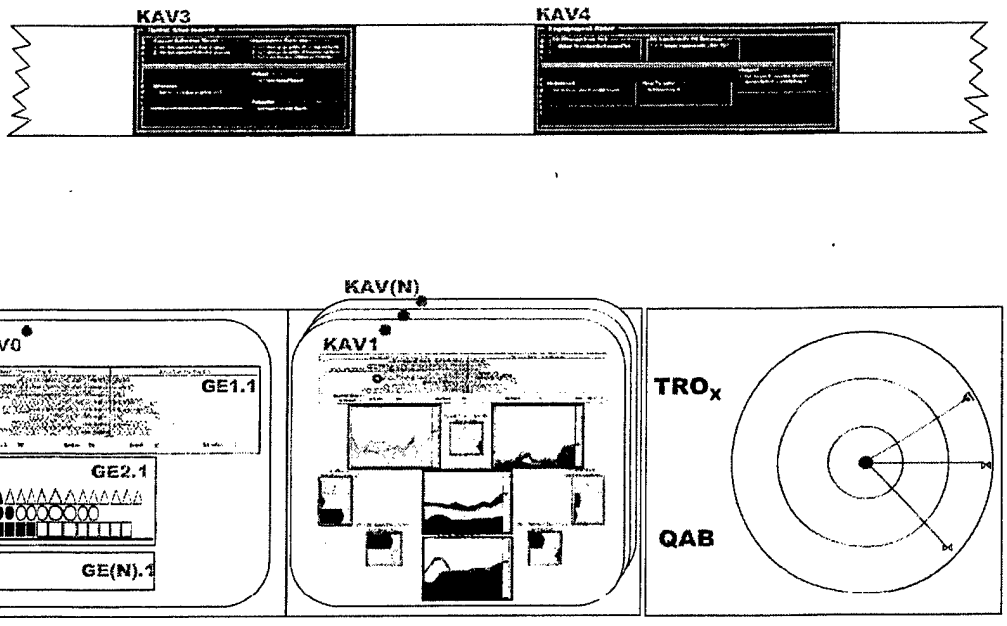

Figure 15. Example of a Physical Workspace Design

D.2.1.2 Interaction Needs

D.2.1.2.1 Presentation Design Concepts Tab

The *Presentation Design Concepts* tab allows the cognitive engineer to quickly assemble PDC diagrams using predefined object types. Selecting this tab will bring up a scrollable view port that will permit the construction of the PDC objects for inclusion in the project's decision aiding system. The tools available for use via the **Tool Pane** will contain each of the types of objects (i.e., controls, axis, control marks, presentation graphics, data entry, etc) that need to be created to construct a PDC. The *Presentation Design Concepts* tab will permit the CSE designer to create a series of AIE objects and relate them to one another, in order to develop aggregate PDC objects. The PDC objects created and managed in this tab then can be associated with other AIE objects. Specifically, the CSE designer will be able to associate RDR, CWR and IRR objects with the PDC object. By extension, these associations will map the PDC objects onto FAN objects, which translates into the PDC object's functional coverage.

Each of the following actions will generate an AIE object. These objects will interact with one another within the *Presentation Design Concepts* tab, as well as with AIE objects in other panes. The following is a partial list of what the CSE designer will be able to within the *Presentation Design Concepts* tab of the **Designer Main Display Pane**:

- Create Presentation objects
- Specify Presentation object type (i.e., bar chart; line graph, etc.)
- Specify Axes for PDC objects
- Specify Scale for PDC objects
- Specify Control Marks for PDC objects
- Specify Data Mark types for PDC objects
- Aggregate PDC objects
- Import graphics from other packages
- Annotate imported graphics with PDC objects
- Associate PDC objects with RDR objects
- Associate PDC objects with CWR and IRR objects

D.2.1.2.1.1 Local Automation

The *Presentation Design Concepts* tab provides some automatic services for the CSE designer. The creation of some AIE objects within the *Presentation Design Concepts* tab will lead to the creation of other objects within the for *Presentation Design Concepts* tab. The following is a partial list of what the AIE system will provide the following automated support within the *Presentation Design Concepts* tab of the **Designer Main Display Pane**:

- Number PDC objects sequentially based on their association with Window objects
- Create default Axes on creation of Presentation object
- Alert CSE designer for the need to associate a PDC object with RDR, CWR and IRR objects on creation
- Highlight local impact of selection of associated AIE objects made in other panes

D.2.1.2.1.2 Interaction with Other Panes

The creation of AIE objects within the *Presentation Design Concepts* tab automatically leads to the creation of AIE objects in other panes. The following is a partial list of the automatic features that the AIE

system will provide when AIE objects are manipulated within the *Presentation Design Concepts* tab of the **Designer Main Display Pane**:

- Create default CWR & IRR objects on the creation of a PDC object
- Modify PDC objects based on changes made to Commodity object
- Create default FAN object on the creation of a PDC object
- Create default Display Layout object on the creation of a PDC object
- Create default Navigation object on the creation of a PDC object
- Highlight location of selected PDC object within **Navigation Overview Pane**
- Highlight location of selected PDC object within **Display Layout Overview Pane**

Actions within other AIE panes will impact AIE objects within the *Presentation Design Concepts* tab of the **Designer Main Display Pane.** This is only a partial list of the features that the AIE system will provide.

- Instantiate Template imported from Pattern Library with appropriate PDC objects
- Create a PDC objects for orphan Window objects
- Create a PDC objects for orphan Navigation objects

D.2.1.2.2 Display Layout Design Tab

The *Display Layout Design* tab allows the cognitive engineer to quickly assemble diagrams for display and pop-up windows using predefined object types. Selecting this tab will bring up a scrollable view port that will permit the construction of the Display Layout objects for inclusion in the project's decision aiding system. The **Tool Pane** will display those tools that are necessary to create a Display Layout object when the *Display Layout Design* tab is selected. The *Display Layout Design* tab will permit the CSE designer to create a series of AIE objects and relate them to one another, in order to develop complete display or pop-up window. Specifically the CSE designer will be able to compose a Display Layout object of PDC objects and other AIE objects. By extension, these associations will map the Display Layout objects onto FAN objects, which translates into the display or pop-up window's functional coverage.

Each of the following actions will generate an AIE object. These objects will interact with one another within the *Display Layout Design* tab, as well as with AIE objects in other panes. The following is a partial list of what the CSE designer will be able to within the *Display Layout Design* tab of the **Designer Main Display Pane**:

- Create Menu Bar objects
- Create System Overview Navigation objects
- Create Window Control objects
- Select PDC objects for inclusion in a Display Layout object
- Organize Display Layout objects into a single display or pop-up Window object

D.2.1.2.2.1 Local Automation

The *Display Layout Design* tab provides some automatic services for the CSE designer. The creation of some AIE objects within the *Display Layout Design* tab will lead to the creation of other objects within the for *Display Layout Design* tab. The following is a partial list of what the AIE system will provide the following automated support within the *Display Layout Design* tab of the **Designer Main Display Pane**:

- Number Windows sequentially based on order of creation
- Create default Menu Bar object on Window object creation
- Create default System Overview Navigation object on Window object creation

- Create default Window Control object on Window object creation
- Alert CSE designer for the need to enter a Display Purpose explanation paragraph on Window object creation
- Highlight local impact of selection of associated AIE objects made in other panes

D.2.1.2.2.2 Interaction with Other Panes

The creation of AIE objects within the *Display Layout Design* tab automatically leads to the creation of AIE objects in other panes. The following is a partial list of the automatic features that the AIE system will provide when AIE objects are manipulated within the *Display Layout Design* tab of the **Designer Main Display Pane**:

- Create default Navigation Node object on the creation of a Window object
- Modify PDC objects based on changes made to Commodity object
- Create default PDC object on the creation of a Window object
- Create default Navigation Link object on the creation of a Window Control object

Actions within other AIE panes will impact AIE objects within the *Display Layout Design* tab of the **Designer Main Display Pane.** This is only a partial list of the features that the AIE system will provide.

- Instantiate Template imported from Pattern Library with appropriate Display Layout objects
- Create a Window objects for orphan Navigation objects

D.2.1.2.3 Navigation Diagram Tab

The *Navigation Diagram* tab allows the cognitive engineer to quickly assemble diagrams for linking defined display and pop-up windows. Selecting this tab will bring up a scrollable view port that will permit the construction of the Navigation Scheme for the project's decision aiding system. The **Tool Pane** will display those tools that are necessary to create Navigation objects when the *Navigation Diagram* tab is selected. The *Navigation Diagram* tab will permit the CSE designer to specify the type of a Window object, the connections between Window objects, and the actions necessary to move from Window Object to another.

Each of the following actions will generate an AIE object. These objects will interact with one another within the *Navigation Diagram* tab, as well as with AIE objects in other panes. The following is a partial list of what the CSE designer will be able to within the *Navigation Diagram* tab of the **Designer Main Display Pane**:

- Create Navigation Link objects
- Specify Window object type
- Create Presentation Device object
- Organize Window objects into a Navigation Scheme
- Display functional coverage of displays using FAN as organizing scheme
- Display functional coverage of displays using Navigation Scheme as organizing scheme

D.2.1.2.3.1 Local Automation

The *Navigation Diagram* tab provides limited automatic services for the CSE designer. The following is a partial list of what the AIE system will provide the following automated support within the *Navigation Diagram* tab of the **Designer Main Display Pane**:

- Alert CSE designer for the need to enter an explanation paragraph on Navigation Link object creation

- Highlight local impact of selection of associated AIE objects made in other panes

D.2.1.2.3.2 Interaction with Other Panes

The following is a partial list of the automatic features that the AIE system will provide when AIE objects are manipulated within the *Presentation Design Concepts* tab of the **Designer Main Display Pane**:

- Create default Window object on the creation of a Navigation Node object
- Create default Window Control object of creation of a Navigation Link object

D.2.1.2.4 Physical Workspace Design Tab

The *Physical Workspace Design* tab allows the cognitive engineer to quickly assemble diagrams for where the windows developed within virtual information will reside within the constraints of the physical workspace. While the virtual workspace defines the different information displays that are available based on the output of the ACWA™ methodology, the Physical Workspace must take into account assumptions about the physical components/construction of the space, expected IT infrastructure upgrades, crew composition, etc. Selecting this tab will bring up a scrollable view port that will permit the construction of the Navigation Scheme for the project's decision aiding system. The **Tool Pane** will display those tools that are necessary to create a 2-dimensional Physical Workspace Layout, when the *Physical Workspace Design* tab is selected. The *Physical Workspace Design* tab will provide the CSE designer the ability to illustrate the position of Presentation and Input devices relative to other equipment/features of the physical workspace.

Each of the following actions will generate an AIE object. These objects will interact with one another within the *Physical Workspace Design* tab, as well as with AIE objects in other panes. The following is a partial list of what the CSE designer will be able to within the *Physical Workspace Design* tab of the **Designer Main Display Pane**:

- Create Presentation Device objects
- Create Input Device objects
- Create Equipment objects
- Create Wall/Barrier objects
- Create other Work Environmental objects (i.e., chairs, desks, etc.)

D.2.1.2.4.1 Local Automation

The *Physical Workspace Design* tab provides only limited automatic services for the cognitive engineer. Devices designated as part of the project's decision aiding system will be automatically numbered, based on their order of entry. All other objects in the physical workspace will use an alternate numbering scheme. Project relevant devices will be displayed to reflect their relevance. Objects added to the physical workspace diagram will be appropriated scaled based on template features.

D.2.1.2.4.2 Interaction with Other Panes

The *Physical Workspace Design* tab has limited interaction with the other panes in the AIE system. Each Presentation Device object will be associated with a Navigation Scheme object. Therefore, the creation of one of the objects will necessitate the object. However, the deletion of a Presentation Device object will not impact the corresponding Navigation Scheme object.

D.2.2 RDR Pane

The **RDR Pane** is a scrolling pane, which contains a tree-like component for displaying representation design requirements (RDR). Initially, all RDR appear as a list. As RDR objects are associated with particular PDC objects and Window Objects, they are placed into a tree structure. The **RDR Pane** is primarily used for the entry, display, and management of text.

D.2.2.1 Design Objective

The purpose of the RDR Pane is to provide the CSE designer a workspace to simultaneously present RDR objects while developing and modifying PDC objects and Display Layout objects. The **RDR Pane** is a repository to transform the CWR and IRR objects developed during analysis into design-oriented objects. This transformation becomes the RDR objects. This AIE object can then be associated with other AIE objects within the project. Figure 16 provides an example of RDRs.

RDR 2.1-1	Provide representation of the sequence and temporal characteristics of decision points (branch & sequels) available for the current plan
RDR 2.1-2	Show "state' of decision point [Active (in current plan) with Default Selected; Active with Alternate Selected (Branch only)]
RDR 2.1-3	Show type of decision point [Branch; Sequel]
RDR 2.1-4	Provide decision point name on rollover
RDR 2.1-5	Show earliest planned time for decision point with respect to associated actions and targets
RDR 2.1-6	Provide means for navigating to Decision Point Map display
RDR 2.1-7	Provide means for navigating to Decision Point Detail pop-up
RDR 2.1-8	Provide indication of multiple decision points at single point in time

Figure 16. Example Representation Design Requirements

D.2.2.2 Interaction Needs

The primary input for cognitive engineers in the **RDR Pane** will be text. RDR objects can be entered either directly by the CSE designer, or they can be auto-generated by the input of other AIE objects. The tools available for use via the **Tool Pane** will include those necessary for simple text editing functions, as well as tools necessary for creating RDR objects. The CSE designer will be able to link RDR objects to other AIE objects within the project. The **RDR Pane** also will support the expansion and contraction of RDR object's associations within the tree. The following is a partial list of the actions the **RDR Pane** will enable:

- Create AIE object association between RDR objects and CWR objects
- Create AIE object association between RDR objects and IRR objects
- Create AIE object association between RDR objects and PDC objects
- Create AIE object association between RDR objects and Display Layout objects

D.2.2.2.1 Local Automation

The **RDR Pane** provides some automatic services for the CSE designer. The AIE system will provide the following automated support within the **RDR Pane**:

- Number RDR objects sequentially based on their association with PDC and Display Layout objects
- Highlight Commodity objects
- Highlight local impact of selection of associated AIE objects made in other panes

D.2.2.2.2 Interaction with Other Panes

The creation of AIE objects in the **RDR Pane** automatically leads to the creation of AIE objects in other panes. The following is a partial list of the automatic features that the AIE system will provide when AIE objects are manipulated within the **RDR Pane**:

- Create default Goal-Process Node on the creation of orphan RDR objects
- Create default PDC object on the creation of RDR objects
- Create default Display Layout object on the creation of RDR objects
- Create default Navigation object on the creation of RDR objects

Actions within other AIE panes will impact AIE objects within the **RDR Pane.** Again, this is only a partial list of the features that the AIE system will provide.

- Create default RDR objects on the creation of a Goal-Process Node
- Create default RDR objects on the creation of a Link between Goal-Process Nodes
- Create default RDR objects on the creation of Flow Model nodes
- Modify RDR objects based on changes made to Commodity object
- Create RDR objects for orphan PDC objects

D.2.3 Navigation Overview Pane

The **Navigation Overview Pane** is a scrolling pane that displays the project's complete navigation space. It will possess tabs, if there are multiple navigation schemes for the project, due to multiple presentation devices. The **Navigation Overview Pane** is used only for the presentation of context information. The **Navigation Overview Pane** always shows the entire navigations scheme for the selected presentation device. In addition, the **Navigation Overview Pane** highlights the region of the navigation scheme that is currently the focus of the **Designers Main Display Pane**.

D.2.3.1 Design Objective

The purpose of this pane is to provide the CSE designer with a means to support movement between the displays/components that will comprise the final decision aiding system. This feature becomes more important as the number of displays increases within the project.

D.2.3.2 Interaction Needs

The network diagram portrayed in the **Navigation Overview Pane** represents all of the displays that are part of the whole decision aiding system. Clicking on any one of the nodes in the network diagram will take the CSE designer to that display in the **Designers Main Display Pane**. Clicking on nodes in the **Navigation Overview Pane** is the only interaction that the CSE designer has with this pane.

D.2.3.2.1 Local Automation

Based on the selected window object or PDC object the **Navigation Overview Pane**, the program will auto scroll to that node and highlight it. No other automation occurs within this pane.

D.2.3.2.2 Interaction with Other Panes

As noted above, selecting one of the nodes in the **Navigation Overview Pane** changes what is displayed in the **Designers Main Display Pane**. No other interaction with the AIE panes or AIE objects is expected.

D.2.4 The Display Layout Overview Pane

The **Display Layout Overview Pane** is only active when the *Display Layout Design* tab or the *Presentation Design Concepts* tab has been selected within the **Designers Main Display Pane**. The **Display Layout Overview Pane** is a scrolling pane that displays the location of selected Window objects. It will possess tabs, if there are multiple Windows for the project. The **Display Layout Overview Pane** is used only for the presentation of context information. In addition, the **Display Layout Overview Pane** highlights the PDC object that is currently the focus of the **Designers Main Display Pane**.

D.2.4.1 Design Objective

The purpose of this pane is to provide the CSE designer a means to support movement between the PDC objects that will comprise the final decision aiding system. This feature becomes more important as the number of PDC objects increases within the project.

D.2.4.2 Interaction Needs

The diagram portrayed in the **Display Layout Overview Pane** represents all of the PDC objects that are part of the selected window object. Clicking on any one of the components in the diagram will take the CSE designer to that PDC object in the **Designers Main Display Pane**. Clicking on components in the **Display Layout Overview Pane** is the only interaction that the CSE designer has with this pane.

D.2.4.2.1 Local Automation

Based on the selected PDC object the **Display Layout Overview Pane**, the program will auto scroll to that node and highlight it. No other automation occurs within this pane.

D.2.4.2.2 Interaction with Other Panes

As noted above, selecting one of the components in the **Display Layout Overview Pane** changes what is displayed in the **Designers Main Display Pane**. No other interaction with the AIE panes or AIE objects is expected.

D.2.5 Salience Pane

The **Salience Pane** is a scrollable pane with two linked columns. This pane is only active when the *Display Layout Design* tab is selected. The first column is list of all of the components for a selected Window object. This component list is derived from the PDC objects and other Display Layout objects that comprise the selected Window. The second column contains the user defined salience values for the components. The **Salience Pane** also contains context region marking to the CSE designer in making a salience assessment. An example of a salience map for a display can be seen in Figure 17.

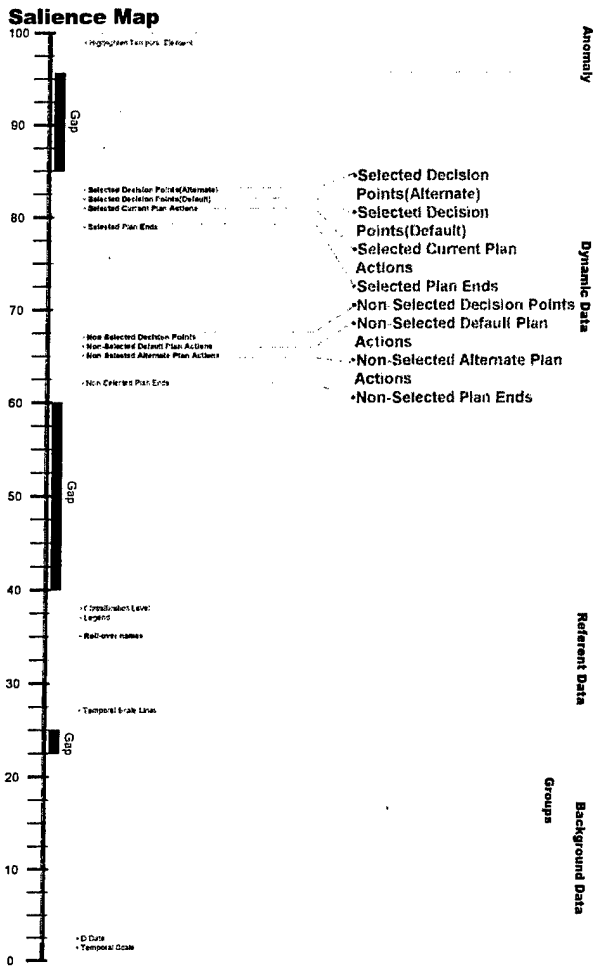

Figure 17. Example of a Salience Map

D.2.5.1 Design Objective

The purpose of the **Salience Pane** is to allow the CSE developer to specify the salience values for the individual components that were used to construct a display or pop-up window. Explicit management of salience by the CSE designer is necessary to ensure that the appropriate window components receive the most attention by users of the project's decision aiding system.

D.2.5.2 Interaction Needs

The primary input for CSE designer in the **Salience Pane** will be text. The tools available for use via the **Tool Pane** will be similar to those used in most other simple text editors. The CSE analyst will be able to modify the names of the component objects in the **Salience Pane**. The CSE designer will not be able to add components in this pane. All components will have to be added in the *Display Layout Design* tab or the *Presentation Design Concepts* tab. The CSE designer also will be able to enter salience values for each of the components of the selected display.

D.2.5.2.1 Local Automation

The **Salience Pane** provides only limited automatic services for the cognitive engineer. The AIE system automatically adds the components to the list and assigns a salience value of zero for all display

components on their creation. The **Salience Pane** will order components first, based on their salience value, and second, alphabetically by their name.

D.2.5.2.2 Interaction with Other Panes

Selecting **Salience Pane** entries will cause the highlighting of associated AIE objects. No other interaction with the AIE panes or AIE objects is expected.

D.3 Project Management Panes

In the virtual workspace design, the *Project Management* panes are located on the top and bottom of the workspace. Figure 18 highlights the panes that are discussed in this section.

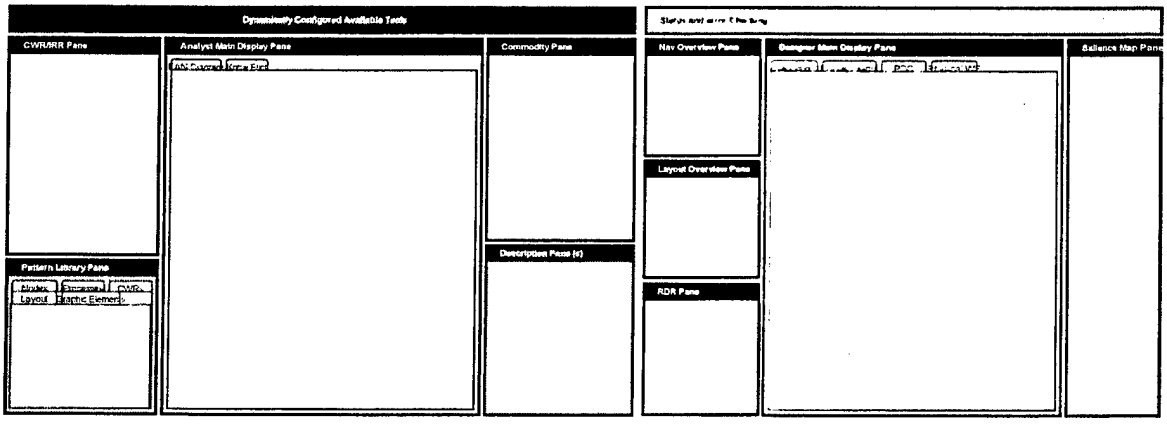

Figure 18. Project Management Panes Location Within AIE's Workspace Design

D.3.1 Object Attributes Pane

The **Object Attribute Pane,** labeled in the figure as the "Description Pane," is a scrolling pane that displays detailed information about the selected object within one of the other panes. This pane will have several views, depending on the AIE object being selected. At least three potential views seem to be possible, given the previously described panes within the AIE system: (1) an expanded text description of the selected AIE object, (2) an attribute-value table of the selected AIE object, or (3) an explanation for how to use a feature/component of the AIE system. While this pane is primarily for displaying information about AIE objects, it also will be possible for the cognitive engineer to edit the fields in the **Object Attribute Pane**.

D.3.1.1 Design Objective

The purpose of the **Object Attribute Pane** is to provide clarification about individual AIE objects selected within the project. When an AIE object is selected, this pane will display all recorded information about the AIE object.

For some AIE objects (e.g., Goal Process Nodes, Support Supported Links, etc), a description helps clarify and expand on the AIE object's name or label. This expanded description ensures there is no confusion between cognitive engineers about what a specific AIE object represents. Figure 19 represents an example of what the view in the **Object Attribute Pane** would look like for an AIE object that have descriptions associated with it.

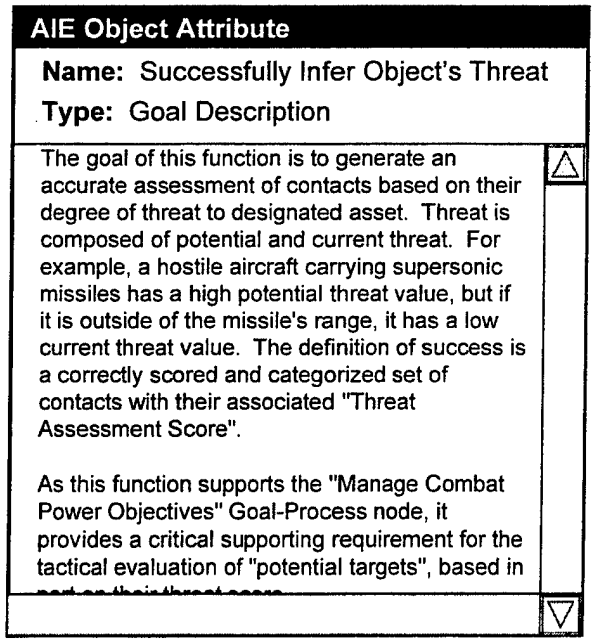

Figure 19. Example of Object Attribute Pane with Text Description

For other AIE objects (e.g., Windows, Navigation Links, etc), an attribute-value table is necessary to specify the AIE object in sufficient detail that it can be coded in software. This expanded attribute-value description ensures there is no confusion between the cognitive engineers and software about how a specific AIE object should behave. Figure 20 represents an example of what the view in the **Object Attribute Pane** would look like for an AIE object that have attribute-value information associated with it.

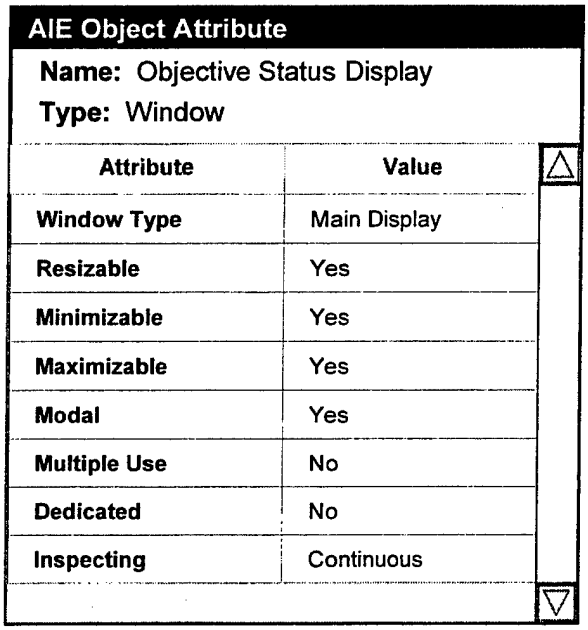

Figure 20. Example of Object Attribute Pane with Attribute Value Table

The **Object Attribute Pane** also will provide information to the cognitive engineer about how components within the AIE system are to be used. The purpose of the **Object Attribute Pane** is as an on-

line context dependent help feature in those cases where a component of the AIE system is touched. Unlike the previous two views into the **Object Attribute Pane,** in this view the cognitive engineer will not be able to edit anything in the pane.

D.3.1.2 Interaction Needs

The **Object Attribute Pane** is primarily a presentation screen. However, under some conditions, the cognitive engineer will be able to edit the information that is displayed. On AIE object creation, the information will need to be developed and input by the cognitive engineer. The tools available for use via the **Tool Pane** will be similar to those used in most other simple text editors. The user will be able to scroll about the **Object Attribute Pane** information without impacting the previously selected AIE object or AIE system component.

D.3.1.2.1 Local Automation

The **Object Attribute Pane** provides only limited automatic services for the cognitive engineer. Definitions that contained references to Commodities will display the Commodity text different than the surrounding text.

D.3.1.2.2 Interaction with Other Panes

Selecting the **Object Attribute Pane** will not cause the highlighting of any AIE objects besides the selected object. Clicking within the **Object Attribute Pane** will change the tools displayed within the **Tool Pane.**

D.3.2 Tool Pane

The **Tool Pane** will provide quick and convenient access to a set of frequently used commands or options, based on the AIE system pane previously selected. The **Tool Pane** will contain a set of buttons that the cognitive engineer can use to perform the task appropriate for any particular AIE system component. An example of the types of buttons that might be available within the **Tool Pane** for a text editable AIE system component can be seen in Figure 21.

Figure 21. Example of Tools Available for Text Editing

D.3.2.1 Design Objective

The purpose of the **Tool Pane** is to support cognitive engineers as they develop AIE objects for the project. The **Tool Pane** should enable the cognitive engineer to quickly find the tools needed to develop the ACWA™ methodological artifacts, which need to be constructed. By changing the tool set based on the type of pane selected, the **Tool Pane** also serves a Help-function. This prevents the cognitive engineer from developing malformed AIE objects.

D.3.2.2 Interaction Needs

The tool icons displayed in the **Tool Pane** will be selectable by the cognitive engineer. Selecting a tool will change the cursor's icon. On mouse over, the tool icons will display their names and the keyboard shortcut for their uses.

D.3.2.2.1 Local Automation

Based on the selected AIE object, the **Tool Pane** will change the type of tool icons displayed. No other automation occurs within this pane.

D.3.2.2.2 Interaction with Other Panes

As noted above, selecting an AIE object in one of the other panes changes what is displayed in the **Tool Pane**. No other interaction with the AIE panes or AIE objects is expected.

D.3.3 Status and Error Checking Pane

The **Status and Error Checking Pane** displays the currently selected AIE object's name, as well as any available information about its approval state, and any information about errors. Errors are generated when an AIE object does not conform to the rule set loaded at AIE system start. A cognitive engineer can disrupt the ACWA™ analysis-design thread and orphan an AIE object, which also will generate an error.

D.3.3.1 Design Objective

The purpose of the **Status and Error Checking Pane** is to support cognitive engineers as they develop AIE objects for the project. If an error message appears, the **Status and Error Checking Pane** should provide an explanation to the cognitive engineer as to how the ACWA™ methodological rules had been violated. The **Status and Error Checking Pane** also provides management oversight capabilities that ensure that the ACWA™ process has been followed.

D.3.3.2 Interaction Needs

Most cognitive engineers using this system will not be able to interact with the **Status and Error Checking Pane**. Only designated reviewers will be able to change the approval status of AIE objects. If the AIE object that generated the error message is modified appropriately, the error message will be removed

D.3.3.2.1 Local Automation

The **Status and Error Checking Pane** will auto-number errors, based on the order of their occurrence. The **Status and Error Checking Pane** will also monitor AIE object approval state, recording which cognitive engineer created the object, which cognitive engineer approved it, as well as relevant temporal information. No other automation occurs within this pane.

D.3.3.2.2 Interaction with Other Panes

As noted above, selecting an AIE object in one of the other panes changes what is displayed in the **Status and Error Checking Pane**. No other interaction with the AIE panes or AIE objects is expected.

D.3.4 Pattern Library Pane

The **Pattern Library Pane** is a tabbed pane containing five tabs that provide the cognitive systems engineer access to templates for various stages in the ACWA™ process. These templates are meant to foster reuse and quicken the system development process. The initial AIE system design envisions five pattern types:

- Goal-Process Node Templates,
- Process Model Templates,
- Cognitive Work Requirement Templates,
- Display Layout Templates, and
- Graphic Element Templates

D.3.4.1 Design Objective

The purpose of this **Pattern Library Pane** is to provide cognitive engineers with a set of templates upon which they can build. Rather than having to create all the AIE objects for a new project from scratch, the cognitive engineer can draw on a library of common patterns that have been used on prior projects and adapt them to the current context. The ability to draw upon prior work will greatly reduce decision aid development time. As the AIE system is used, the holdings in the Pattern Library will continue to grow, reducing the development time even further. The templates contained with the **Pattern Library Pane** are organized by AIE object type to further facilitate its utility.

The *Goal-Process Node Templates* will contain assemblies of goal process nodes that are commonly found in domains, which the ACWA™ methodology has been applied. For example, many work domains contain transportation requirements. There is a typical arrangement of Goal-Process nodes that can be used to functionally satisfy that requirement. The **Pattern Library Pane** will not only provide the default FAN objects for the template, but also the CWR and IRR objects, as well as other down stream AIE design objects.

The *Process Model Templates* will contain a set of standard process models that can be used to quickly populate Goal-Process nodes within the FAN. The process models contained within the **Pattern Library Pane** will be instantiated with default values, when placed into the **Analyst Main Display Pane**. Once placed in the **Analyst Main Display Pane**. These standard process models will be editable, in order to tailor them to the specific features of the project domain.

Similarly, the *Cognitive Work Requirement Templates* will contain a set of frequently used CWR that the cognitive engineer can modify to address the demands of the current work domain. Like the other templates, this one is both a means to save time in completing the ACWA™ methodology for a project, and a potential training tool for novice cognitive engineers.

This *Display Layout Templates* will contain a set of standard display layouts that can be used to quickly populate a window. The purpose of the *Display Layout Templates* is to support the quicker development of visualizations for a system. Once placed in the **Designer Main Display Pane,** these standard layouts will be editable, in order to tailor them to the specific features of the project domain.

This *Graphic Element Template* will contain a set of reusable graphic elements that can be imported into the **PDC Pane** and used to develop an initial mock-up of a system display/pop-up window. The graphic elements contained in this template pane will contain a default set of interaction features, as well as a list of standard CWR and IRR objects.

D.3.4.2 Interaction Needs

The **Pattern Library Pane** will have only limited interaction features. Cognitive engineers will be able to scroll through the various template types and select ones that they are interested in. Cognitive engineers also will be able to drag example templates to the appropriate AIE system pane.

D.3.4.2.1 Local Automation

On mouse over, the templates will display their name. No other automation occurs within this pane.

D.3.4.2.2 Interaction with Other Panes

The selection of template in the **Pattern Library Pane** only will impact the **Object Attribute Pane**. On selection of a template, the **Object Attribute Pane** will provide a description of it, its use, and the work domain for which it was originally created.

E PROPOSED FOLLOW ON WORK

E.1 AIE Phase II: Produce AIE v0.1 Prototype

Based on the user requirements analysis conducted for this report and the Technical Survey Results (ManTech Aegis Research Corporation, 2003), it is possible to envision Phase II in the development of the ACWA Integrated Environment (AIE). Namely, extend what was learned during the technical investigation of Integrated Development Environments (IDE) and use the preliminary Cognitive Systems Engineering Design Specification (CSEDS) developed in this report to construct a limited scope AIE prototype focused on the core CWA analytical artifacts of the ACWA approach. The proposed initial prototype (v0.1) would focus on developing support for the analytical portion of ACWA. AIE version 0.1 prototype will have limited functionality and will not attempt a full implementation of the ACWA methodology. Instead, the AIE version 0.1 prototype will focus on the portion of the ACWA methodology that provides the critical foundation for decision support system development.

Specifically, the v0.1 prototype will focus on constructing AIE system features that will support (1) the development and management of a project's Functional Abstraction Network (FAN); (2) the development, modification, and organization of Cognitive Work Requirements (CWR) and linking of these CWRs to objects that compose the FAN; and (3) the identification and association of Information Relationship Requirements (IRR) necessary for the satisfaction of the CWRs along with the integration features to demonstrate the initial 'linked point of view' characteristics of an IDE. Opportunistically the AIE v0.1 also will include other features to support the Analytic portion of CSE (e.g., development of appropriate holdings for the Pattern Library).

At the completion of the AIE version 0.1 effort, it is expected that a cognitive systems engineer will be able to construct a minimally capable FAN. This FAN will contain all of the components identified in the ACWA methodology (Elm et al, 2003). In addition this tool will auto-generate FAN, CWR and IRR objects within the prototype based on the cognitive engineer's actions and a set of ACWA methodology rules. For example, if a cognitive engineer were to add an unassociated CWR object, AIE v0.1 would add a linked FAN object and a linked IRR object. The prototype AIE also will also support cross pane (integrated) selection; that is, the selection of an object in one pane will cause the selection of associated objects in all panes.

The focus of the Phase II development will be on the demonstration of IDE capabilities within the AIE system. Therefore, the emphasis will be placed on software development rather than interface design. Therefore while AIE v0.1 will be "demo-able," it will not be sufficiently robust that it could be released for use to cognitive engineers outside of the project. For example, this first step will not include a rigorous testing program, development of online help and other support features. These will be deferred to later development phases.

E.2 AIE Phase III: At a crossroads of fielding or functionality

Following the successful development of AIE v0.1 in Phase II, the Phase III direction is variable depending on the needs of the project sponsor's area of interest and desired level of effort.

- One direction of Phase III is to extend the functionality developed in the AIE v0.1 prototype to include features that support the Knowledge Elicitation and Designer portions of ACWA. This path will increase the number of panes included in the AIE system, but not the overall robustness of the software.

- An alternative direction would be to AIE v0.1 and transform it into a fully functioning v1.0 AIE system with Analytic support for cognitive systems engineers. This path would permit the sponsors to utilize a part scope AIE system within their own organization.
- A third potential direction would be to simultaneously undertake development on both of the previous two options. This path would lead to the completion of a robust, fully functioning AIE system quicker, however it will require a greater level of effort and corresponding budget.

Whichever path is chosen, having an ACWA Integrated Development Environment is critical to achieving AFRL/HE's mission of designing decision support systems from a decision centered perspective. Lacking such a tool makes the labor of any such effort unaffordable for mainstream system development programs. The proposed AIE system is a new, powerful, development environment that can revolutionize the design of powerful decision support systems.